



Enterprise Architect User Guide

Enterprise Architect is an intuitive, flexible and powerful UML analysis and design tool for building robust and maintainable software. From requirements gathering, through analysis, modeling, implementation and testing to deployment and maintenance, Enterprise Architect is a fast, feature-rich, multi-user UML modeling tool, driving the long-term success of your software project.



Copyright © 1998-2011 Sparx Systems Pty Ltd

Enterprise Architect User Guide

Introduction

by Geoffrey Sparks

Enterprise Architect is a complete UML-based solution for analysing, designing, managing, sharing and building software systems.

Enterprise Architect User Guide

© 1998-2011 Sparx Systems Pty Ltd

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: November 2011

Publisher

Sparx Systems

Managing Editor

Geoffrey Sparks

Technical Editors

Geoffrey Sparks

Dermot O'Bryan

Simon McNeilly

Neil Capey

Sam Mancarella

John Redfern

Vimal Kumar

Howard Britten

Brad Maxwell

Steve Meagher

Special thanks to:

All the people who have contributed suggestions, examples, bug reports and assistance in the development of Enterprise Architect. The task of developing and maintaining this tool has been greatly enhanced by their contribution.

Table of Contents

Foreword	1
Part I Introduction	3
Overview	5
What You Can Do	7
Key Benefits	9
Key Features	14
Enterprise Architect Editions	20
The Trial Version	20
Editions Available	22
The Read-only 'Lite' Edition	26
Formal Statements	27
Copyright Notice	27
End User License Agreement	28
Trademarks	31
Acknowledgements	34
Order Enterprise Architect	35
Installation	36
Register a Full License	38
Help and Support	40
Available Helpfile Formats	41
Support	42
Part II Getting Started	44
Common Tasks	45
Basics	47
A Quickstart Tutorial	49
Create Project Scenario	49
Add a View To Your Model	51
Add Packages	52
Creating Diagrams	52
Creating New Elements on Diagrams	53
Adding Connectors	55
Modify Properties	56
Moving Elements, Diagrams and Connectors	57
Move Elements Within a Package.....	58
Move Elements Between Packages.....	58
Move Elements in a Diagram.....	59
Move Elements Between Diagrams.....	59
Connect a Different Element on a Diagram.....	62
Delete Elements	62
Remove Elements From a Diagram.....	63
Delete Elements From Your Model.....	64
Delete Connectors	65
Save Changes	67
Create Requirements	67
Requirement Trace Connector	70
Typical Project Roles	71
Summary of Typical Tasks	72

Business Analysts	74
Software Architects	75
Software Engineers	77
Developers	78
Project Managers	80
Testers	80
Implementation Managers	81
Technology Developers	82
Database Developers	84
User Interface Guide	86
Start Page	87
Remove Recent Projects.....	89
Learning Center	89
Main Menu	90
File	92
Edit	93
Paste Elements Submenu.....	94
View	95
View Submenus	98
Project.....	100
Documentation Submenu.....	101
Model Validation Submenu.....	102
Import/Export Submenu.....	102
Version Control Submenu.....	103
Security Submenu.....	104
Diagram.....	105
Diagram Advanced Menu.....	106
Element.....	108
Advanced Submenu.....	109
Inline Features Submenu.....	110
Source Code Engineering Submenu.....	111
Appearance Submenu.....	112
Position Submenus.....	112
Tools.....	115
Source Code Engineering Submenu.....	116
Database Engineering Submenu.....	117
Web Services Submenu.....	118
XML Schema Submenu.....	118
Model Transformation (MDA) Submenu.....	118
Spelling Tools.....	119
Data Management Submenu	119
Manage .EAP File Submenu.....	120
Analyzer.....	120
Debug Menu.....	122
Record Menu.....	123
Test Menu	124
More Analyzer Window s.....	125
Extensions.....	125
Settings	127
Project Types Submenu.....	129
Window	129
Help	131
Standard Windows	132
Code Editors.....	134
Dock Window s.....	135
Autohide Window s.....	137
Standard Toolbars	138
Default Tools Toolbar	139

Project Toolbar	141
Code Generation Toolbar	142
UML Elements Toolbar	143
Diagram Toolbar	145
Current Element Toolbar	146
Current Connector Toolbar	148
Workspace Layouts	149
Debug & Record Toolbar	150
Status Bar	152
Customization	153
The Customize Dialog	153
Customize Commands	154
Customize Toolbars	155
Custom Tools	156
Open External Tools	157
Pass Parameters to Applications	159
Customize Keyboard	160
Customize Menu	161
Customize Options	162
Visual Styles	162
Manage Workspace Layout	163
Other Windows	164
The Output Window	164
The Web Browser	165
Keyboard Shortcuts	166
Keyboard-Mouse Shortcuts	174

Part III Projects and Teams

176

Introduction	178
Projects Defined	179
Open a Project	181
Project Shortcuts	183
Create Project Shortcut	185
Capture Current Work Environment	187
Encrypt Repository Password	188
File Based Repositories	189
Copy a Base Project	190
Copy Existing Project	191
Server Based Repositories	193
Create a Repository	194
Access 2007 Repository	194
MySQL Repository	195
SQL Server Repository	195
Oracle Data Repository	197
PostgreSQL Repository	197
Adaptive Server Anywhere Repository	198
MSDE Server Repository	198
Progress OpenEdge Repository	199
Set Up an ODBC Driver for a Connection to a Repository	200
MySQL ODBC Driver	201
PostgreSQL ODBC Driver	202
ASA ODBC Driver	205
Progress OpenEdge ODBC Driver	206
Oracle ODBC Driver	208
Connect to a Data Repository	209
MySQL Data Repository	210
SQL Server Data Repository	212

Oracle Data Repository (ODBC).....	214
Oracle Data Repository (OLE DB).....	216
PostgreSQL Data Repository	218
ASA Data Repository.....	219
MSDE Server Data Repository	221
Progress OpenEdge Repository.....	221
Upsizing	224
Upsize to MySQL.....	224
Upsize to Access 2007.....	227
Upsize to SQL Server	228
Upsize to Oracle 9i, 10g or 11g.....	230
Upsize to PostgreSQL.....	232
Upsize to Sybase ASA	234
Upsize to MSDE.....	236
Upsize to Progress OpenEdge.....	236
The WAN Optimizer	238
Team Development	240
Project Sharing	240
Share Enterprise Architect Projects.....	242
Refresh View of Shared Project.....	242
Share Projects on Network Drive.....	243
Distributed Development.....	243
Replication.....	244
Design Masters.....	246
Create Replicas.....	246
Synchronize Replicas.....	247
Remove Replication.....	248
Upgrade Replicas.....	249
Resolve Conflicts.....	250
Configure User Security	251
Enable Security	253
Security Policy.....	255
Maintain Users.....	256
Import User IDs From Active Directory.....	257
Assign User To Groups	260
Set Up Single Permissions.....	260
View All User Permissions.....	261
Maintain Groups	262
Set Group Permissions.....	263
List of Available Permissions	264
View and Manage Locks	268
Password Encryption.....	269
Change Password.....	270
Lock Model Elements.....	271
Add Connectors To Locked Elements.....	273
Lock Packages.....	273
Apply a User Lock.....	274
Locked Element Indicators.....	276
Identify Who Has Locked An Object.....	277
Manage Your Own Locks.....	278
Team Review Tools	279
Context Menu	280
Add a New Category	284
Add a New Topic.....	285
Add a New Post.....	287
Reply to a Post.....	289
Edit an Item.....	291
Add Object Links.....	292

Team Review Resources.....	294
Search Team Review.....	295
Team Review Options.....	296
Team Review Connections.....	296
Workflow Scripts.....	298
Workflow Script Functions.....	299
Functions - Validate and Control User Input.....	300
Functions - Create a Search With User Tasks.....	301
Filled Workflow Data Structures.....	301
Workflow Data Structures You Fill.....	303
Functions You Call.....	304
Sharing Reference Data.....	305
Export Reference Data.....	306
Import Reference Data.....	308
Change Management.....	311
Version Control.....	312
Introduction.....	313
Version Control Usage.....	315
Version Control of Model Data.....	316
Version Control and Reference data.....	317
Version Control Basics.....	318
Applying Version Control in a Team Environment.....	319
Version Control Nested Packages.....	320
Project Browser Indicators.....	321
Offline Version Control.....	322
Version Control Branching.....	324
Version Control Product Setup.....	324
System Requirements.....	325
Create a Subversion Environment.....	327
Create a new Repository Sub-tree.....	329
Create a Local Working Copy.....	330
Verify the SVN Workspace.....	331
Subversion Under Wine-Crossover.....	332
Preparing a Subversion Environment Under Wine.....	332
TortoiseSVN.....	334
Create a CVS Environment.....	335
Prepare a CVS Local Workspace.....	336
Verify the CVS Workspace.....	337
TortoiseCVS.....	338
Create a TFS Environment.....	339
TFS Workspaces.....	340
TFS Exclusive Check Outs.....	342
Create an SCC Environment.....	342
Upgrade at Enterprise Architect Version 4.5, Under SCC Version Control.....	344
Version Control Setup.....	344
Re-use an Existing Configuration.....	346
Version Control Settings Dialog.....	346
SCC Settings.....	348
CVS Settings.....	350
SVN Settings.....	351
TFS Settings.....	352
Use Version Control.....	353
Project Version Control Menu.....	355
Package Version Control Menu.....	357
Configure Controlled Package.....	362
Check In a Model Branch.....	364
Check Out a Model Branch.....	365

Check In a Package.....	365
Check Out a Package.....	366
Undo Check Out of a Package.....	367
Update to the Latest Revision of Selected Package.....	368
Update to the Latest Revision of All Packages.....	369
Include Other Users' Packages.....	370
Apply Version Control To Branches.....	371
Export Controlled Model Branch.....	371
Import Controlled Model Branch.....	372
Manually Locating Model Branch Files.....	374
Review Package History.....	376
Review Package History - SCC Client.....	377
Retrieve Prior Revision - SCC Client.....	378
Validate Package Configurations.....	379
Resynchronize the Status of Version Controlled Packages.....	380
Tracking Changes	381
Auditing.....	382
Auditing Quickstart.....	383
Auditing Settings.....	384
The Audit View	387
Audit View Controls.....	389
Audit History Tab.....	392
Auditing Performance Issues.....	393
Audit View Performance Issues.....	394
Package Baselines.....	394
Baselines	396
Manage Baselines.....	398
Create Baselines.....	400
The Compare Utility (Diff).....	401
Compare Options	402
Check Visual Changes to Diagrams.....	403
Example Comparison.....	406
Compare Utility Tab Options.....	408
Model Transfer	411
XML Import and Export.....	411
Export to XML.....	413
Publish Model Package.....	414
Import from XML.....	417
Import EMX/UML2 Files.....	419
Import a Rhapsody Model.....	420
Limitations of XML.....	421
The UML DTD.....	422
Controlled Packages	422
Controlled Package Menu.....	424
Configure Packages.....	426
Remove Package from Control.....	428
Save a Package.....	428
Load a Package	429
Batch XML Export.....	430
Batch XML Import.....	431
Manual Version Control with XML.....	432
Report Deletion of Cross Package References.....	433
CSV Import and Export.....	435
CSV Specifications.....	435
Using Preserve Hierarchy.....	438
CSV Export.....	439
CSV Import.....	440
Perform a Project Data Transfer	442

Copy Packages Between Projects	444
Compare Projects	445
Project Management	447
The Project Management Window	448
Project Resources	449
Resource Allocation.....	450
Effort Management.....	452
Risk Management.....	453
Metrics.....	455
Effort Types	456
Metric Types.....	457
Risk Types.....	458
The Project Information Window	460
Project Tasks.....	460
Add, Modify and Delete Tasks.....	461
Project Issues.....	462
Add, Delete and Modify Issues.....	463
Report From Project Issues Dialog.....	465
Report From Project Issues Tab.....	466
Report Output Sample.....	466
Project Glossary.....	467
Project Glossary Tab.....	468
The Glossary Dialog.....	469
Generate a Report.....	471
Glossary Report Output Sample.....	472
Project Task Allocation	472
Resource View	473
Element View	475
Report View	477
Spell Checking	480
Select Spell Checker Options.....	481
Use Languages Other Than English.....	483
Using the Spell Checker	485
Correcting Words.....	486
Personal Information	486
Internal Mail.....	487
Create a Message.....	492
Recording Work.....	493
Monitor Your Tasks.....	496
Monitor Workflow	498
Working Sets.....	498
Project Calendar	502
Calendar.....	505
Configure Event Subtypes.....	509
Allocated Resources.....	510
Project Tasks.....	513
Use Case Estimation	516
Technical Complexity Factors	517
Environment Complexity Factors.....	518
Default Hours	520
Estimating Project Size.....	520
SPEM	523
SPEM Toolbox Pages.....	524
Update Package Status	529
Manage Bookmarks	530
Monitor Change Events	532
Project Maintenance	534
Check Project Data Integrity	535

Reset Table Auto Increment or Identity Columns	536
Project Upgrade	537
Upgrade Replicas.....	538
Run SQL Patches	539
Rename a Project	539
Compact a Project	539
Repair a Project	540
Local Options	542
General	544
Standard Colors	546
Create Project Colors.....	548
Diagram	549
Appearance.....	550
Set Default Fonts.....	552
Behavior.....	553
Sequence.....	555
Objects	556
Element Visibility.....	559
Links	559
Communication Message Colors	561
XML Specifications	562

Part IV Navigate, Search and Trace 566

Navigate: Exploring Your Model	567
The Project Browser	567
Project Browser Context Menus.....	569
Model (Root Node) Context Menu.....	569
Package Menu.....	571
Add Sub-Menu.....	574
Documentation Sub-Menu.....	575
Code Engineering Sub-Menu.....	575
Execution Analyzer Sub-Menu.....	576
Import/Export Sub-Menu.....	577
Contents Sub-Menu.....	577
Element Menu - Project Browser.....	577
Add Sub Menu.....	580
Diagram Menu - Project Browser.....	581
Operation Menu - Project Browser.....	582
Project Browser Toolbar.....	583
Project Browser Icon Overlays.....	585
Order Package Contents.....	586
Set Default Behavior.....	587
Package Browser	587
List Header.....	590
Package Browser Options.....	592
Diagram List	596
Model Views	598
Model Views Toolbar.....	600
Model Views Context Menus.....	601
Model Views Operations.....	604
Diagram Slide Show.....	607
The Pan & Zoom Window	610
Search: Finding Information	611
Model Search	611
Model Search Context Menu.....	614
Model Search Toolbar.....	616
Pre-defined Searches.....	617

Customizing the Search View	619
Create & Modify Searches	621
Create Search Definitions	623
Advanced Search Options	626
Add Filters	627
Fields and Conditions	629
Diagram Filters	629
Work With Diagram Filters	631
Trace: Tracking Dependencies	635
Traceability Tools	636
The Traceability Window	637
Relationship Matrix	639
Open the Relationship Matrix	640
Set Element Type	640
Set Connector Type and Direction	641
Set Source and Target Package	641
Matrix Profiles	643
Creating and Deleting Relationships	644
Relationship Matrix Options	645
Review Source and Target Elements	647
The Relationships Window	648
Example Diagram	649
Gap Analysis Matrix	651
Edit Gap Notes	654

Part V Modeling Basics **657**

Modeling	659
Models	660
Model Wizard	660
Model Templates	662
Business Process Model Template	665
Requirements Model Template	666
Use Case Model Template	666
Domain Model Template	667
Class Model Template	668
Database Model Template	669
Component Model Template	670
Deployment Model Template	671
Testing Model Template	672
Maintenance Model Template	673
Project Model Template	674
Views	676
Add Views	677
Rename Views	678
Delete Views	678
Packages	680
Open Package in the Project Browser	680
Add a Package	681
Rename a Package	682
Copy a Package	683
Drag a Package Onto a Diagram	684
Show or Hide Package Contents	685
Delete a Package	685
Diagrams	687
Diagram Context Menu	687
Insert Elements and Connectors	690

Print Preview	690
Diagram View	691
Format Toolbar	692
Diagram Tabs	695
View Tab Sw itcher.....	697
Diagram Toolbox	698
Toolbox Appearance Options	701
Toolbox Shortcut Menu.....	704
Common Group.....	706
Use Case Group.....	707
Class Group.....	708
Object Group.....	709
Composite Group.....	709
Communication Group.....	710
Interaction Group.....	710
Timing Group.....	711
State Group.....	711
Activity Group.....	712
Component Group.....	713
Deployment Group.....	713
Profile Group.....	714
Metamodel Group.....	715
Analysis Group.....	715
Custom Group.....	716
Requirement Group.....	717
Maintenance Group.....	717
User Interface Group.....	718
WSDL Group.....	719
XML Schema Group.....	720
Data Modeling Group.....	720
Test Domain Group.....	721
Diagram Tasks	721
Add New Diagrams.....	724
Diagram Properties.....	725
General Tab.....	726
Diagram Tab.....	727
Elements Tab.....	729
Features Tab.....	731
Visible Class Members.....	732
Connectors Tab.....	732
Paste from Project Brow ser.....	733
Paste Multiple Items.....	735
Paste Composite Elements.....	736
Paste Activities	736
Copy And Paste Diagram Element.....	737
Place Related Elements on Diagram.....	738
Delete Diagram.....	738
Rename Diagram.....	739
Change Diagram Type.....	740
Diagram Navigation Hotkeys.....	740
Copy Image to Disk.....	741
Copy Image to Clipboard.....	742
Copy (Duplicate) Diagram.....	742
Z Order Elements	744
Set the Default Diagram.....	744
Open Package From Diagram.....	744
Feature Visibility.....	745
Insert Diagram Properties Note.....	747

Create Legends.....	748
Legend Properties.....	749
Style Options.....	750
Autosize Elements.....	750
Swimlanes.....	751
Swimlane Details.....	753
Swimlanes Matrix.....	753
Using the Image Manager.....	755
Create Custom Diagram Background.....	757
Import Image Library.....	758
Show Realized Interfaces of Class.....	760
Label Menu Section.....	760
Pan and Zoom a Diagram.....	763
Move Elements In Diagram Sections.....	763
View Last and Next Diagram.....	764
Set Diagram Page Size.....	764
Scale Image to Page Size.....	766
Lock Diagram.....	767
Undo Last Action.....	768
Redo Last Action.....	768
Layout Diagrams	769
Circular/Elliptical Layout.....	770
Box Layout.....	775
Per Page Layout.....	776
Digraph Layout.....	778
Spring Layout.....	780
Neaten Layout.....	781
Fan Relations Layout.....	783
Converge/Diverge Layout.....	784
Auto Route Layout.....	786
Lay Out a Diagram Automatically	787
The Quick Linker	791
Create New Elements.....	792
Create Connectors.....	793
Elements	795
Element Tasks	795
Create Elements.....	797
Add Elements Directly To Packages.....	798
Use Auto Naming and Auto Counters	799
Set Element Parent.....	801
Show Element Use.....	802
Set Up Cross References.....	803
Move Elements Within Diagrams.....	805
Move Elements Between Packages.....	806
Copy Elements Between Packages	808
Change Element Type.....	809
Align Elements.....	809
Resize Elements.....	810
Delete Elements.....	811
Customize Visibility of Elements.....	813
Create Notes and Text.....	814
Link Note to Internal Documentation.....	815
Set an Element's Default Appearance.....	817
Get/Set Project Custom Colors.....	819
Set Element Templates Package.....	820
Highlight Context Element.....	822
Make Linked Element a Local Copy	823
Copy Features Between Elements.....	823

Move Features Between Elements	824
Composite Elements	825
Show Embedded Elements In Composite.....	826
Element Context Menu	827
Properties Menu Section.....	828
Advanced Submenu.....	829
Add Submenu.....	832
Insert Related Elements.....	833
Find Submenu.....	833
Embedded Elements Submenu.....	834
Embedded Elements Window.....	835
Features Menu Section.....	836
Code Engineering Menu Section.....	836
Appearance Menu Section.....	837
Set Element Font.....	839
Element Multiple Selection Menu.....	840
Visual Representation	841
Element Icons.....	841
Compartment.....	842
Element Property Displays	843
Properties Dialog.....	843
General Settings.....	845
Details.....	846
Advanced Properties.....	847
Templates.....	848
Requirements.....	848
External Requirements.....	850
Constraints.....	850
Links.....	851
Scenarios.....	852
Structured Specification Tab.....	854
Set Up Scenario Specification.....	855
Structured Specification Toolbar.....	857
Structured Specification Item Context Menu.....	859
Structured Specification Selected Text Context Menu.....	861
Structured Specification Entry Points Context Menu.....	863
Structured Specification Floating Toolbar.....	864
Generate Diagrams.....	864
Generated Activity Diagram.....	865
Generated RuleFlow Diagram.....	867
Generated State Machine Diagram.....	867
Generated Sequence Diagram.....	869
Generated Robustness Diagram.....	871
Generate Scenario From Activity Diagram.....	873
Generate Test Cases.....	873
Context References Tab.....	874
Scenario Constraints Tab.....	875
Associated Files.....	875
The Element Browser.....	876
Properties Window.....	878
The Scenarios & Requirements Window.....	879
Select <Item> Dialog.....	880
Select Property Dialog.....	882
Set Feature Dialog.....	884
Attributes	885
Attributes Dialog - General.....	887
Attributes Dialog - Detail.....	889
Attributes Dialog - Constraints.....	890

Attribute Tagged Values	891
Create Properties	892
C# Automatic Property Support.....	893
Display Inherited Attributes	895
Create Object From Attribute.....	896
Object Classifiers	898
Using Classifiers	899
Behavior	899
Operations.....	900
Operations Dialog - General.....	901
Operations Dialog - Behavior.....	904
Initial Code.....	906
Operations Dialog - Constraints.....	906
Operation Tagged Values.....	907
Override Parent Operations.....	908
Display Inherited Operations.....	908
Interactions and Activities.....	910
Behavior Calls	910
Associate with Different Behaviors	911
Synchronize Arguments.....	911
Behavior Call Arguments.....	912
Behavior Parameters.....	912
Parameters Dialog.....	913
Parameter Tagged Values	914
Operation Parameters by Reference.....	915
In-place Editing Options	916
In-place Editing Tasks.....	916
Edit Element Item Name.....	918
Edit Feature Stereotype.....	919
Edit Feature Scope.....	920
Edit Attribute Keyw ord.....	921
Edit Operation Parameter Keyw ord.....	922
Edit Parameter Kind.....	923
Insert New Feature	924
Insert Operation Parameter	924
Insert Maintenance Feature.....	925
Insert Testing Features	926
Linked Documents	928
Create Document Artifact.....	929
Link Document to UML Element.....	929
Edit Linked Documents	930
Hyperlink From Linked Document	931
Create Element From Document.....	932
Replace or Delete Documents.....	933
Create Linked Document Templates.....	934
Edit Linked Document Templates.....	934
Connectors	936
Connector Context Menu	936
Properties Menu Section.....	937
Type-Specific Menu Section.....	938
Advanced Menu Section.....	939
Style Menu Section.....	940
Appearance Menu Section.....	941
Connector Tasks	942
Connect Elements.....	943
Connect to Element Feature.....	944
Add a Note to a Connector.....	946
Arrange Connectors	946

Change Connector Type.....	947
Change the Source or Target Element.....	947
Connector Styles.....	948
Create Connector in Project Browser.....	952
Relationship Visibility.....	953
Delete Connectors.....	954
Generalization Sets.....	955
Hide/Show Connectors.....	955
Hide/Show Labels.....	957
Connector In-place Editing Options.....	957
Reverse Connector.....	958
Set Association Specializations.....	958
Change Sequence Message Scope.....	959
Show Uses Arrow Head.....	959
Tree Style Hierarchy.....	960
Connector Properties	961
Connector Constraints.....	963
Binding.....	964
Source Role.....	964
Target Role.....	967
Connector Tagged Values.....	967
Connector Advanced.....	968
Tagged Values	969
Quick Add - Tagged Value To Elements	971
Assign a Tagged Value to an Item	972
Assign Notes to a Tagged Value	973
Modify Tagged Values	973
Show Duplicate Tags	974
Advanced Tag Management	975
Notes	977
Notes Toolbar	979
Reference Data	981
UML Types	981
Stereotype Settings.....	982
Shape Editor.....	984
Tagged Value Types.....	985
Cardinality.....	986
Data Types.....	987
People	989
Project Authors.....	989
Select Users.....	990
Project Roles.....	991
Project Resources.....	992
Project Clients.....	993
General Types	995
Status Types.....	995
Constraint Types.....	997
Constraint Status Types.....	998
Requirement Types.....	998
Scenario Types.....	999
Metrics and Estimation.....	1000
Maintenance	1001
Problem Types.....	1001
Testing Types.....	1002
Resources	1003
Favorites.....	1005

Part VI Standard UML Models**1008**

UML Diagrams	1010
UML Structural Models	1011
Package Diagram	1011
Example Package Diagram.....	1013
Class Diagram	1014
Example Class Diagram.....	1015
Object Diagram	1015
Example Object Diagram.....	1016
Composite Structure Diagram	1017
Example Composite Structure Diagram.....	1018
Properties.....	1019
Deployment Diagram	1020
Example Deployment Diagram.....	1021
Component Diagram	1023
Example Component Diagram.....	1024
Profile Diagram	1025
Example Profile Diagram.....	1026
UML Behavioral Models	1027
Activity Diagram	1028
Example Activity Diagram.....	1029
Use Case Diagram	1030
Example Use Case Diagram.....	1032
State Machines	1032
Example State Machine.....	1034
Pseudo-States.....	1037
Regions.....	1037
State Machine Table	1038
State Machine Table Options.....	1039
State Machine Table Operations	1042
Change State Machine Table Position.....	1042
Change State Machine Table Size.....	1043
Insert New State.....	1043
Insert Trigger.....	1044
Insert/Change Transition.....	1045
Reposition State or Trigger Cells.....	1046
Add Legend.....	1046
Find Cell in State Machine Diagram.....	1047
State Machine Table Conventions.....	1047
Export State Table To CSV File.....	1048
Example State-Trigger Table.....	1049
Example State-Next State Table.....	1050
Timing Diagram	1051
Example Timing Diagram.....	1052
Create a Timing Diagram.....	1053
Set a Time Range.....	1053
Edit a Timing Diagram.....	1054
Add and Edit State Lifeline.....	1054
Add States to a State Lifeline.....	1055
Edit States in a State Lifeline.....	1056
Delete States in a State Lifeline.....	1057
Edit Transitions In State Lifeline.....	1057
Add and Move Transitions.....	1058
Add and Edit Value Lifeline.....	1060
Add States In Value Lifeline.....	1061
Edit Transitions In Value Lifeline.....	1061

Configure Timeline - States.....	1062
Numeric Range Generator.....	1064
Configure Timeline - Transitions.....	1065
Time Intervals.....	1066
Create Time Intervals.....	1067
Compress Time Intervals.....	1069
Select Time Intervals.....	1070
Time Interval Operations.....	1071
Sequence Diagram.....	1074
Example Sequence Diagram.....	1076
Denote Lifecycle of an Element.....	1076
Layout of Sequence Diagrams.....	1077
Sequence Elements.....	1078
Sequence Diagrams and Version Control.....	1079
Sequence Element Activation.....	1080
Lifeline Activation Levels.....	1082
Sequence Message Label Visibility.....	1083
Change the Top Margin.....	1083
Inline Sequence Elements.....	1084
Communication Diagram.....	1084
Example Communication Diagram.....	1085
Communication Diagrams in Color.....	1086
Interaction Overview Diagram.....	1087
Example Interaction Overview Diagram.....	1088
UML Elements.....	1090
Behavioral Diagram Elements.....	1090
Action.....	1091
Local Pre/Post Conditions.....	1093
Action Notation.....	1095
Action Pin.....	1097
Assign Action Pins.....	1098
Action Expansion Node.....	1099
Class Operations in Activity Diagrams.....	1100
Activity.....	1101
Activity Notation.....	1103
Activity Parameter Nodes.....	1103
Activity Partition.....	1105
Actor.....	1107
Central Buffer Node.....	1107
Choice.....	1108
Combined Fragment.....	1110
Create a Combined Fragment.....	1112
Interaction Operators.....	1113
Datastore.....	1116
Decision.....	1117
Diagram Frame.....	1119
Diagram Gate.....	1120
Endpoint.....	1121
Entry Point.....	1123
Exception.....	1123
Expansion Region.....	1124
Exit Point.....	1126
Final.....	1127
Flow Final.....	1128
Fork/Join.....	1130
Fork.....	1131
Join.....	1133
History.....	1134

Initial.....	1136
Interaction.....	1137
Interaction Occurrence.....	1139
Interruptible Activity Region.....	1141
Junction.....	1142
Lifeline.....	1144
Merge.....	1145
Message Endpoint.....	1145
Message Label.....	1146
Note.....	1147
Partition.....	1148
Receive.....	1149
Region.....	1150
Send.....	1151
State.....	1152
Composite State.....	1153
State/Continuation.....	1155
Continuation.....	1155
State Invariant.....	1157
State Lifeline.....	1158
State Machine.....	1161
Structured Activity.....	1161
Structured Node.....	1163
Sequential Node.....	1164
Loop Node.....	1164
Conditional Node.....	1168
Synch.....	1169
System Boundary.....	1170
Boundary Element Settings.....	1171
Terminate.....	1172
Trigger.....	1173
Use Case.....	1174
Use Case Extension Points.....	1176
Rectangle Notation.....	1177
Value Lifeline.....	1177
Structural Diagram Elements	1179
Artifact.....	1179
Create Artifact For External File.....	1180
Class.....	1181
Active Classes.....	1182
Parameterized Classes (Templates).....	1183
Collaboration.....	1184
Collaboration Use.....	1186
Component.....	1187
Data Type.....	1188
Deployment Spec.....	1189
Device.....	1190
Document Artifact.....	1191
Enumeration.....	1191
Execution Environment.....	1192
Expose Interface.....	1193
Information Item.....	1194
Interface.....	1194
Node.....	1196
Object.....	1196
Run-time State.....	1197
Define a Run-time Variable.....	1198
Remove a Defined Variable.....	1198

Object State.....	1199
Package.....	1200
Part.....	1200
Add Property Value.....	1201
Port.....	1202
Add a Port to an Element.....	1203
Inherited and Redefined Ports.....	1203
The Property Page.....	1204
Primitive.....	1204
Signal.....	1205
UML Connectors	1206
Aggregation	1208
Change Aggregation Connector Form.....	1209
Assembly	1209
Association	1210
Qualifiers.....	1211
Qualifiers Dialog.....	1212
Association Class	1215
Connect New Class to Association.....	1216
Communication Path	1217
Composition	1217
Connector	1218
Control Flow	1219
Delegate	1220
Dependency	1221
Apply a Stereotype.....	1221
Deployment	1222
Extend	1223
Generalization	1224
Include	1224
Information Flow	1225
Using Information Flow s.....	1226
Convey Information on a Flow	1228
Realize an Information Flow	1229
Interrupt Flow	1230
Manifest	1231
Message	1231
Message (Sequence Diagram).....	1232
Self-Message.....	1235
Call	1236
Message Examples.....	1237
Change the Timing Details.....	1238
General Ordering.....	1240
Asynchronous Signal Message.....	1241
Co-Region Notation.....	1242
Message (Communication Diagram).....	1242
Create a Communication Message.....	1243
Re-Order Messages	1244
Message (Timing Diagram).....	1245
Create a Timing Message.....	1246
Nesting	1249
Notelink	1249
Object Flow	1250
Object Flow s in Activity Diagrams	1250
Occurrence	1252
Package Import	1252
Package Merge	1253
Realization	1254

Recursion	1255
Role Binding	1255
Represents	1256
Representation	1257
Template Binding	1257
Parameter Substitution	1258
Trace	1260
Transition	1260
Use	1263
UML Stereotypes	1264
Apply Stereotypes	1265
Stereotype Selector	1266
Stereotype Visibility	1266
Standard Stereotypes	1267
Stereotypes with Alternative Images	1269
Design Patterns	1271
Create a Pattern	1271
Import a Pattern	1273
Use a Pattern	1274
Add Pattern Dialog	1276
UML Profiles	1277
Import a UML Profile	1279
Add Profile Objects to a Diagram	1280
Tagged Values in Profiles	1280
Synchronize Tagged Values and Constraints	1281
MDG Technologies	1283
Work with MDG Technologies	1284
Manage MDG Technologies	1285
Access Remote MDG Technologies	1287
Import MDG Technologies to APPDATA	1288
Extensions - MDG Technologies	1289
Define a Modeling Language	1292
MDG Technology SDK	1292
Developing Profiles	1293
Custom Stereotypes	1293
Create Profiles	1295
Create a Profile Package	1296
Add Stereotypes and Metaclasses	1297
Define Stereotype Tagged Values	1299
With Predefined Tag Types	1300
With Supported Attributes	1301
Use the Tagged Value Connector	1301
Define Stereotype Constraints	1302
Add Enumeration Elements	1303
Add Shape Scripts	1304
Set Default Appearance	1306
Export a Profile	1306
Save Profile Options	1307
Supported Attributes	1308
Define a Stereotype as a Metatype	1312
Define Multiple-Stereotype Level	1313
Define Creation of Instance	1313
Create Composite Elements	1314
Define Child Diagram Types	1315
Stereotype Profiles	1316
Quick Linker	1317

Quick Linker Definition Format.....	1318
Quick Linker Example.....	1320
Hide Default Quick Linker Settings.....	1322
Quick Linker Object Names.....	1322
MDG Technologies - Creating.....	1323
Create MDG Technologies.....	1324
Add a Profile.....	1327
Add a Pattern.....	1328
Add a Diagram Profile.....	1329
Add a Toolbox Profile.....	1329
Add Learning Center Pages.....	1330
Add Tagged Value Types.....	1331
Add Code Modules.....	1331
Add MDA Transforms.....	1333
Add Images.....	1333
Add Scripts.....	1334
Add RTF Report Templates.....	1334
Add Linked Document Templates.....	1335
Working with MTS Files.....	1335
Customize Toolbox Profiles.....	1337
Create Toolbox Profiles.....	1337
Toolbox Page Attributes.....	1339
Add Custom Compartments to Element.....	1339
Create Hidden Sub-Menus.....	1341
Override Default Toolboxes.....	1342
Assign Icons To Toolbox Items.....	1342
List of Enterprise Architect Toolboxes.....	1343
Elements Used in Toolboxes.....	1344
Connectors Used In Toolboxes.....	1345
Create Custom Diagram Profiles.....	1346
Built-In Diagram Types.....	1348
Attribute Values - StyleEx & pdata.....	1348
Create Learning Center Profiles.....	1350
Define Learning Center Toolboxes.....	1350
Define Links To Documents.....	1351
Built-In Learning Center Commands.....	1352
Run Add-In Functions.....	1353
Save a Learning Center Profile.....	1354
Define Validation Configuration.....	1354
Incorporate Model Templates.....	1355
Set Up Technology Element Images.....	1356
Deploy An MDG Technology.....	1357
Shape Scripts.....	1358
Getting Started With Shape Scripts.....	1358
Shape Editor.....	1359
Shape Script Overview.....	1360
Write Scripts.....	1363
Shape Attributes.....	1363
Drawing Methods.....	1366
Color Queries.....	1372
Conditional Branching.....	1373
Query Methods.....	1373
Display Element/Connector Properties.....	1373
Sub-Shapes.....	1376
Reserved Names.....	1377
Miscellaneous.....	1378
Syntax Grammar.....	1379
Example Scripts.....	1380

Tagged Value Types.....	1385
Predefined Structured Types.....	1386
Create Structured Tagged Values.....	1389
Predefined Reference Data Types.....	1390
Create Reference Data Tagged Values.....	1392
Create Custom Tagged Value Type.....	1393
Code Template Framework.....	1394
Code and Transform Templates.....	1395
Base Templates.....	1397
Export Code Generation and Transformation Templates.....	1400
Import Code Generation and Transformation Templates.....	1401
Synchronize Code.....	1401
Synchronize Existing Sections.....	1403
Add New Sections.....	1403
Add New Features and Elements.....	1404
The Code Template Editor.....	1404
Code Template Syntax.....	1406
Literal Text.....	1406
Variables.....	1407
Macros.....	1409
Template Substitution Macros.....	1411
Field Substitution Macros.....	1411
Substitution Examples.....	1413
Attribute Field Substitution Macros.....	1414
Class Field Substitution Macros.....	1416
Code Generation Option Field Substitution Macros.....	1418
Connector Field Substitution Macros.....	1423
Constraint Field Substitution Macros.....	1426
Effort Field Substitution Macros.....	1427
File Field Substitution Macros.....	1427
File Import Field Substitution Macros.....	1428
Link Field Substitution Macros.....	1429
Linked File Field Substitution Macros.....	1430
Metric Field Substitution Macros.....	1431
Operation Field Substitution Macros.....	1431
Package Field Substitution Macros.....	1432
Parameter Field Substitution Macros.....	1433
Problem Field Substitution Macros.....	1434
Requirement Field Substitution Macros.....	1435
Resource Field Substitution Macros.....	1436
Risk Field Substitution Macros.....	1436
Scenario Field Substitution Macros.....	1437
Tagged Value Substitution Macros.....	1437
Test Field Substitution Macros.....	1438
Function Macros.....	1439
Control Macros.....	1444
List Macro.....	1444
Branching Macros.....	1446
Synchronization Macros.....	1447
The Processing Instruction (PI) Macro.....	1448
EASL Code Generation Macros.....	1449
EASL Collections.....	1451
EASL Properties.....	1454
Call Templates From Templates.....	1461
The Code Template Editor in MDG Development.....	1462
Custom Templates.....	1462
Override Default Templates.....	1464
Add New Stereotyped Templates.....	1464

Create Custom Language Template.....	1466
Part VII Requirement Models	1468
Requirements	1471
Create Requirements	1473
Requirement Properties	1475
Color Code External Requirements	1477
Extend Requirement Properties	1478
Display Tagged Values On Diagrams	1479
Connect Requirements	1480
Import Requirements and Hierarchies in CSV	1481
Model Requirements	1483
Feature	1485
Internal Requirements	1486
Make Internal Requirement External.....	1486
Manage Requirements	1488
View Requirements	1489
Trace Use of Requirements	1490
Manage Requirement Changes	1491
Report on Requirements	1492
Requirements Diagram	1494
Example Requirements Diagram	1494
Part VIII Domain Based Models	1498
Domain Based Diagrams	1502
Analysis Models	1503
Mind Mapping	1503
Custom Diagram	1505
Data Flow Diagrams	1506
Analysis Stereotypes	1508
Analysis Diagram	1509
Example Analysis Diagram.....	1511
Business Models	1512
Business Modeling/Interaction	1514
Example Business Modeling Diagram.....	1514
Business Models	1515
Analysis Models.....	1517
Process Modeling Notation.....	1518
Inputs, Resources and Information.....	1519
Events	1520
Outputs	1521
Goals	1521
A Complete Business Process.....	1522
Business Rules	1522
Model Business Rules For RuleTasks.....	1527
Create a Business Domain Model.....	1529
Create a Rule Flow Model.....	1530
Pass Parameters to Rule Flow Activity.....	1533
Model Rules In an Operation.....	1534
Compose Business Rules	1536
Validate Business Rules	1542
Code Generation For Business Rules.....	1543
BPMN Models	1545
BPMN 1.0 and 1.1 Toolbox Pages.....	1547
BPMN 2.0 Toolbox Pages	1549

BPMN 2.0 Business Process Toolbox Pages.....	1550
BPMN 2.0 Choreography Toolbox Pages.....	1553
BPMN 2.0 Collaboration Toolbox Page.....	1555
BPMN 2.0 Conversation Toolbox Pages.....	1557
BPMN 2.0 Type Toolbox Page.....	1560
Change BPMN Element Appearance.....	1561
Migrate BPMN 1.0 Model to BPMN 1.1.....	1565
Migrate BPMN 1.1 Model to BPMN 2.0.....	1567
BPMN 2.0 XML.....	1569
BPEL Models	1570
BPEL 1.1 Model.....	1571
Create BPEL 1.1 Model Structure.....	1573
Model a BPEL 1.1 Process.....	1576
Start Event.....	1577
Intermediate Event.....	1578
Activity	1581
Gateway.....	1584
End Event.....	1586
Sequence Flow	1587
Pool	1588
Assignment.....	1589
Create BPEL 1.1 Web Service.....	1591
Generate BPEL 1.1.....	1594
BPEL 2.0 Model.....	1595
Create BPEL 2.0 Model Structure.....	1597
Model a BPEL 2.0 Process.....	1599
Start Event.....	1601
Intermediate Event.....	1602
Activity	1605
Gateway.....	1607
End Event.....	1608
Data Object.....	1610
Property	1611
Sequence Flow	1612
Pool	1612
Assignment.....	1613
Create BPEL 2.0 Web Service Operation.....	1615
Generate BPEL 2.0.....	1618
BPEL Model Validation.....	1619
ArchiMate	1621
Eriksson-Penker Extensions	1623
Data Models	1625
Conceptual Data Model	1626
Logical Data Model	1627
Entity Relationship Diagrams (ERDs)	1627
Table	1630
Database Schema	1631
Geodatabase Design for ArcGIS	1633
ArcGIS UML Profile	1633
ArcGIS Toolbox Pages.....	1634
Setting ArcGIS Coordinate Systems.....	1637
Export ArcGIS XML Workspace	1640
Import ArcGIS XML Workspace	1641
MDG Technology for ODM	1643
ODM Toolbox Pages	1644
OWL Elements & Relationships.....	1646
RDF Elements & Relationships.....	1649

Example ODM Diagrams	1651
ODM Commands	1652
User Interface Models	1655
Example User Interface Diagram	1656
Screen	1656
UI Control Elements	1657
Web Stereotypes	1660
Win32 UI Technology	1661
Other Stereotypes	1663
Boundary	1663
Create a Boundary.....	1664
Control	1665
Create a Control Element.....	1665
Entity	1666
Create an Entity.....	1667
Event	1667
Hyperlinks	1668
Hyperlinks To Files	1669
Script Hyperlinks	1670
Add Action As Hyperlink.....	1670
Hyperlinks Between Diagrams.....	1670
Image	1672
N-Ary Association	1673
Process	1674
Packaging Component	1674
Risk	1675
Task	1676
Test Case	1677

Part IX Model Transformation 1679

Transform Elements	1683
Chaining Transformations	1684
Built-in Transformations	1686
C# Transformation	1687
Data Model To ERD Transformation	1689
DDL Transformation	1689
EJB Transformations	1694
ERD To Data Model Transformation	1697
Java Transformation	1699
JUnit Transformation	1702
NUnit Transformation	1704
WSDL Transformation	1705
XSD Transformation	1707
Transformation Templates	1710
Write Transformations	1712
Default Transformation Templates	1713
Intermediary Language	1714
Intermediary Language Debugging	1714
Objects	1716
Connectors	1722
Transform Connectors	1724
Transform Foreign Keys	1726
Copy Information	1727
Convert Types	1728
Convert Names	1728
Cross References	1730

Part X Software Engineering	1733
Development Tools	1734
Overview of Development	1736
Application Patterns (Model + Code)	1738
MDG Integration and Code Engineering	1740
Modeling Conventions	1741
ActionScript Conventions	1742
Ada 2005	1743
C Conventions	1745
Object Oriented Programming In C.....	1747
C# Conventions	1748
C++ Conventions	1750
Managed C++ Conventions	1752
C++/CLI Conventions.....	1753
Delphi Conventions	1754
Java Conventions	1756
AspectJ Conventions.....	1757
PHP Conventions	1758
Python Conventions	1759
System C Conventions	1760
VB.Net Conventions	1761
Verilog Conventions	1763
VHDL Conventions	1764
Visual Basic Conventions	1766
Generate Source Code	1768
Generate a Single Class	1770
Generate a Group of Classes	1771
Generate a Package	1772
Update Package Contents	1774
Synchronize Model and Code.....	1776
Namespaces	1777
Generate From Behavioral Models	1778
SW Code Generation - State Machine Diagrams	1780
Java Code Generated From State Machine Diagram.....	1781
State Machine Modeling For HDLs	1784
Code Generation - Interaction Diagrams	1786
Code Generation - Activity Diagrams	1787
Importing Source Code	1790
Import Source Code	1792
Notes on Source Code Import	1793
Import a Directory Structure	1795
Import Binary Module	1797
Classes Not Found During Import	1798
Editing Source Code	1799
Languages Supported	1800
Compare Editors	1801
Code Editor Toolbar	1802
Code Editor Context Menu	1805
Create Use Case for Method.....	1807
Code Editor Functions	1809
Function Details.....	1810
Intellisense.....	1812
Find and Replace.....	1814
Find in Files.....	1817

Code Editor Key Bindings	1819
Code Build & Debug	1824
Getting Started	1825
Prerequisites	1825
General Workflow	1825
Workspace Layout and Tools	1826
Analyzer Scripts	1827
Managing Analyzer Scripts.....	1828
Analyzer Script Editor	1831
Add Build Commands	1832
Clean	1833
Testing.....	1834
Add Testing Command.....	1834
Testpoints Output.....	1836
Debug.....	1837
Operating System Specific Requirements	1837
UAC-Enabled Operating Systems.....	1838
WINE Debugging.....	1839
Microsoft C++ and Native (C, VB).....	1841
General Setup.....	1841
Debug Symbols.....	1842
Java	1843
General Setup for Java.....	1843
Advanced Techniques.....	1845
Attach to Virtual Machine.....	1845
Internet Browser Java Applets.....	1845
Working with Java Web Servers.....	1846
JBoss Server.....	1848
Apache Tomcat Server.....	1849
Apache Tomcat Windows Service.....	1850
.NET	1850
General Setup for .NET.....	1850
Debugging an Unmanaged Application.....	1851
Debug COM Interop.....	1852
Debug ASP .NET.....	1853
The PHP Debugger.....	1854
PHP Debugger - System Requirements.....	1857
PHP Debugger Checklist.....	1857
The GNU Debugger.....	1859
Tracepoint Output.....	1860
Workbench Setup.....	1861
Add Run Command.....	1862
Add Deploy Command.....	1862
Configure Recording Detail.....	1863
Build Application	1866
Locate Compiler Errors in Code.....	1866
Debugging	1867
Before Starting.....	1867
Breakpoint and Marker Management.....	1869
Setting Code Breakpoints.....	1871
Breakpoint Properties.....	1871
Failure to Bind Breakpoint.....	1873
Trace Statements.....	1873
Debugger Facilities.....	1874
Run the Debugger.....	1875
View the Local Variables	1877
View Content Of Long Strings.....	1879
View Debug Variables in Code Editors.....	1879

View Variables in Other Scopes.....	1880
View Elements of Array.....	1881
Variable Snapshots.....	1881
Break When a Variable Changes Value.....	1882
View the Call Stack.....	1884
Inspect Process Memory.....	1886
Create Sequence Diagram of Call Stack.....	1886
Show Loaded Modules.....	1887
Debug Another Process.....	1888
Process First Chance Exceptions.....	1889
Run a Debug Session.....	1890
Other Settings	1891
Source Code Engineering Options	1891
Source Code Options.....	1892
Import Component Types.....	1894
Options - Code Editors.....	1895
Editor Language Properties.....	1896
Options - Object Lifetimes.....	1898
Options - Attribute/Operations.....	1899
Code Page for Source Editing.....	1900
Local Paths	1900
Local Paths Dialog	1901
Language Macros	1902
Set Collection Classes	1904
Example Use of Collection Classes.....	1906
Language Options	1907
ActionScript Options.....	1909
Ada 2005 Options.....	1909
ArcGIS Options.....	1910
C Options.....	1911
C# Options.....	1912
C++ Options.....	1912
Delphi Options.....	1913
Delphi Properties.....	1914
Java Options.....	1915
PHP Options.....	1916
Python Options.....	1917
SystemC Options.....	1917
VB.NET Options.....	1918
Verilog Options.....	1919
VHDL Options.....	1919
Visual Basic Options.....	1920
MDG Technology Language Options.....	1921
Reset Options.....	1922
Code Template Framework	1924
ICONIX	1925
GoF Patterns	1928
Part XI Database Engineering	1931
Physical Data Model	1932
Create a Data Model Diagram	1933
Example Data Model Diagram.....	1934
Tables and Columns	1935
Create a Table.....	1936
Working with Table Properties.....	1937
Set the Database Type.....	1938

Set Table Owner/Schema.....	1939
Set MySQL Options.....	1939
Set Oracle Table Properties.....	1940
Create Columns.....	1942
Reorder Columns.....	1944
Data Types	1944
DBMS Product Conversion for a Package.....	1945
Data Type Conversion For a Table.....	1946
Add New Datatypes.....	1947
MySQL Data Types.....	1949
Oracle Data Types.....	1949
Map Data Types Between DBMS Products.....	1950
Database Keys	1951
Primary Key.....	1952
Create a Primary Key.....	1953
Define Primary Key Name Template.....	1954
SQL Server Non Clustered Keys.....	1955
Foreign Key.....	1956
Create a Foreign Key.....	1956
Composite Foreign Key.....	1959
Define Foreign Key Name Template.....	1960
Stored Procedures	1961
Create a Stored Procedure.....	1961
Indexes	1963
Create an Index.....	1963
Triggers	1965
Create a Trigger.....	1965
Check Constraints	1966
Create a Check Constraint.....	1966
Views	1967
Create a View.....	1968
Oracle Packages	1970
Supported Databases	1970
Default Constraints	1971
Import Database Schema	1973
Select a Data Source.....	1975
Select Tables.....	1976
The Imported Class Elements.....	1976
Generate DDL	1978
Generate DDL For a Table.....	1978
Generate DDL for a Package.....	1979
Data Modeling Notations	1982

Part XII Systems Engineering 1984

SysML	1990
SysML Activity.....	1992
SysML Block Definition.....	1996
SysML Interaction.....	1999
SysML Internal Block.....	2000
SysML Model Elements.....	2002
SysML Parametrics.....	2005
SysML Requirements.....	2007
SysML State Machine.....	2009
SysML Use Case.....	2011
SysML Parametric Models	2013
Simulate a SysML Model.....	2016

A SysML Requirements Model	2018
A SysML Operational Domain Model	2019
Compose System Design	2021
Create Reusable Subsystems	2023
Migrate SysML 1.1 Model to SysML 1.2	2024
Part XIII SOA and XML	2026
XML Schema - XSD	2027
Model XSD	2027
Schema Package.....	2029
Global Element.....	2031
Local Element.....	2032
Global Attribute.....	2033
Local Attribute.....	2035
Attribute Group.....	2036
Complex Type.....	2037
Model Group.....	2039
Group.....	2040
Simple Type.....	2041
Union.....	2042
Enumeration.....	2043
Any.....	2044
Any Attribute.....	2045
Abstract XSD models	2046
Default UML to XSD Mappings.....	2048
Generate XSD	2049
Generate Global Element.....	2051
Import XSD	2053
Global Element and ComplexType.....	2054
UML Profile for XSD	2055
Profile for XML Schema.....	2056
XSD Datatypes Package.....	2065
Web Services - WSDL	2067
Create WSDL 1.1 Model Structure	2067
Model WSDL	2069
WSDL Namespace.....	2070
WSDL Message.....	2073
WSDL Message Part.....	2074
WSDL Port Type.....	2075
WSDL Port Type Operation.....	2076
WSDL Binding.....	2078
WSDL Binding Operation.....	2080
WSDL Service.....	2082
WSDL Document.....	2083
Generate WSDL	2085
Import WSDL	2086
SoaML	2088
SOMF	2092
Generate MOF	2094
Getting Started	2095
Export MOF to XMI	2097
Part XIV Model Simulation	2100
How It Works	2102

How it Looks	2104
Simulation Windows	2106
Set Up Simulation Script	2109
Activate Simulation Script	2111
Run Model Simulation	2112
Simulation Breakpoints	2114
Dynamic Simulation with Javascript	2116
Guards and Effects	2119
Triggers	2122
Simulation Events Window	2125
Waiting Triggers	2129
Re-Signal Triggers	2130
Trigger Parameters	2131
Trigger Sets and Auto-Firing	2133
Using Trigger Sets to Simulate an Event Sequence	2136
Multi-threading - Forks and Joins	2137
Multi-threading - Concurrent State Regions	2138
BPMN Simulation	2139
Creating a BPMN Simulation Model	2139
Comparing UML Activities to BPMN Processes	2140

Part XV Execution Analyzer 2143

Visual Execution Analyzer Samples	2145
Recording Sequence Diagrams	2147
How it Works	2148
The Recording History.....	2149
Working With Recording History.....	2151
Diagram Features.....	2152
Setup for Recording	2153
Control Stack Depth.....	2153
Place Recording Markers	2154
Set Record Markers	2155
Marker Types	2155
The Breakpoints & Events Window	2157
Working w ith Marker Sets.....	2158
Recording Activity for a Class	2160
Control the Recording Session	2161
Start Recording.....	2161
Step Through Function Calls	2162
Recorder Toolbar	2163
Generating Sequence Diagrams	2164
Reporting State Transitions	2165
Reporting a State Machine.....	2166
Recording and Mapping State Changes.....	2168
Profiling Native Applications	2170
System Requirements	2171
Getting Started	2171
Start & Stop the Profiler	2173
Profiler Operation	2174
Setting Options	2174

Save and Load Reports	2175
Save Report in Team Review	2175
Object Workbench	2176
How it Works	2176
Create & Delete Workbench Instances	2177
Invoke Methods	2179
Unit Testing	2183
Set Up Unit Testing	2183
Run Unit Tests	2185
Record Test Results	2185
Testpoint Management	2187
The Testpoints Window	2189
The Testpoints Window Toolbar	2191
Class Invariant Dialog	2193
Operation Constraints Dialog	2193
Testpoint Editor	2195
Combine Testpoints	2196
Test Cut	2198
Test Set	2198
Test Suite	2199

Part XVI Testing 2202

Model Validation	2203
Configure Model Validation	2205
Rules Reference	2206
Well-Formedness	2207
Element Composition	2208
Property Validity	2208
OCL Conformance	2209
Testing	2211
The Testing Workspace	2213
The Test Details Dialog	2214
Unit Testing	2215
Integration Testing	2216
System Testing	2218
Acceptance Testing	2220
Scenario Testing	2221
Move or Copy Tests Between Categories	2223
Import Scenario as Test	2224
Import Test From Other Elements	2226
Import Responsibility or Constraint as Test	2227
Create Maintenance Item From Test	2228
Show Test Script Compartments	2229
Test Documentation	2230

Part XVII Maintenance 2233

Maintenance Diagram	2234
Example Maintenance Diagram	2235
Maintenance	2236
The Maintenance Workspace	2237
Maintenance Item Properties	2238
Move or Copy Maintenance Items	2240
Create Elements From Maintenance Item	2241
Show Maintenance Script in Diagram	2242
Changes and Defects	2244

Defects (Issues)	2244
Changes	2246
Element Properties	2247
Assign People to Defects or Changes	2247

Part XVIII Reporting 2249

RTF Documents	2251
Generate RTF Documents	2253
Diagram Options	2255
Generate RTF Documentation Dialog	2256
Resource Documents	2259
Exclude Package from Report	2260
Document Options	2261
Element Filters	2264
Other Filters	2267
Project Constants	2268
Word Substitution	2269
Language Substitution	2269
RTF Templates	2270
Document Template Designer	2272
Document Template Designer Options	2275
Scroll Through Text	2276
File and Print Options	2277
Cut and Paste Options	2279
View Options	2281
Image and Object Inserts	2282
Character Formatting	2284
Paragraph Formatting	2285
Tab Support	2288
Page Breaks and Repagination	2289
Headers and Footers	2290
Hyperlinks and Bookmarks	2291
Table Commands	2292
Sections and Columns	2296
Stylesheets and Table of Contents	2297
User-Defined Section Numbering	2299
Frames and Drawing Objects	2302
Search and Replace Commands	2303
Setting Sections for Reporting	2304
Tabular Sections	2305
Child Sections	2307
Embedded Elements Sections	2308
Reporting Linked Documents	2309
Reporting Profiled Relationship Matrices	2310
Constraint and Scenario Sections	2310
Report Elements From External Packages	2312
Tagged Value Sections	2314
Add Section Content	2315
Import RTF Template	2316
The Legacy RTF Report Generator	2318
Document a Single Element	2319
Set the Main RTF Properties	2319
Apply a Filter	2320
Exclude Elements	2321
RTF Diagram Format	2321
Model Include	2321
RTF Report Options	2322
RTF Report Selections	2323

Generate the Report.....	2323
Legacy RTF Style Templates.....	2323
Save as Document.....	2325
Custom Language Settings.....	2326
Virtual Documents	2327
Create Master Document.....	2330
Create Model Document.....	2331
Add Packages to Model Document.....	2333
Delete Package in Model Document.....	2334
Document Order.....	2335
Section Numbering in Virtual Documents.....	2337
Generate the Document (RTF).....	2338
Other Documents	2339
Dependency Report.....	2339
Diagrams Only Report.....	2340
Implementation Report.....	2341
Implementation Targets Dialog.....	2342
Testing Report.....	2342
Testing Details Report.....	2343
Use MS Word	2344
Open a Report in Microsoft Word.....	2345
Change Linked Images to Embedded.....	2345
RTF Bookmarks.....	2346
Other Features of Word.....	2347
Add Table of Contents.....	2348
Add Table of Figures.....	2349
Add Headers and Footers.....	2349
Manipulate Tables in Word.....	2349
Refresh Links.....	2351
HTML Reports	2352
Create an HTML Report	2353
The Generate HTML Report Dialog	2354
Web Style Templates	2355
HTML Template Fragments	2357
Master Documents and Model Documents	2368
Generate a Master Document (HTML).....	2368

Part XIX Automation

2371

Scripting	2372
Scripts Tab	2373
Script Group Properties.....	2375
Console Tab	2376
Script Editor	2378
Enterprise Architect Object Model	2381
Using the Automation Interface	2382
Connect to the Interface.....	2382
Set References In Visual Basic.....	2385
Examples and Tips.....	2386
Call from Enterprise Architect.....	2387
Available Resources.....	2389
Reference	2389
Interface Overview Package.....	2390
Element Features Package.....	2391
Attribute Class.....	2392
AttributeConstraint Class.....	2396
AttributeTag Class.....	2397
CustomProperties Collection.....	2399

EmbeddedElements Collection.....	2399
Method Class	2400
MethodConstraint Class.....	2404
MethodTag Class	2405
Parameter Class.....	2406
ParamTag Class.....	2408
Partitions Collection.....	2409
Properties Class.....	2410
Transitions Collection.....	2412
App Object.....	2412
Enumerations.....	2413
ConstLayoutStyles.....	2414
CreateBaselineFlag.....	2416
CreateModelType.....	2416
DocumentBreak.....	2416
DocumentType.....	2416
EAEditionTypes.....	2417
EnumRelationSetType	2417
ExportPackageXMIFlag.....	2418
MDGMenus.....	2418
MessageFlag.....	2419
ObjectType.....	2419
PropType	2421
ReloadType.....	2421
ScenarioDiagramType.....	2422
ScenarioStepType.....	2422
ScenarioTestType.....	2423
XMIFlag	2423
Repository Package.....	2423
Author Class.....	2425
Client Class	2425
Collection Class.....	2427
Datatype Class.....	2429
EventProperties Class.....	2431
EventProperty Class	2432
ModelWatcher Class.....	2432
Package Class	2433
ProjectIssues Class	2442
ProjectResource Class	2444
ProjectRole Class.....	2445
PropertyType Class	2446
Reference Class.....	2447
Repository Class.....	2448
Stereotype Class	2470
Task Class	2471
Term Class.....	2473
Element Package.....	2474
Constraint Class.....	2476
Effort Class.....	2477
Element Class.....	2478
File Class	2491
Issue (Maintenance) Class	2492
Metric Class.....	2494
Requirement Class.....	2495
Resource Class.....	2497
Risk Class	2498
Scenario Class.....	2499
ScenarioExtension Class.....	2501

ScenarioStep Class	2502
TaggedValue Class	2504
Test Class	2506
Connector Package	2507
Connector Class	2508
ConnectorConstraint Class	2514
ConnectorEnd Class	2515
ConnectorTag Class	2518
RoleTag Class	2520
Diagram Package	2522
Diagram Class	2523
DiagramLinks Class	2528
DiagramObject Class	2529
SwimlaneDef Class	2531
Swimlanes Class	2532
Swimlane Class	2534
Project Interface Package	2534
Project Class	2535
Document Generator Interface Package	2561
DocumentGenerator Class	2562
Mail Interface Package	2566
MailInterface Class	2566
Code Samples	2569
Open the Repository	2570
Iterate Through a .EAP File	2570
Add and Manage Packages	2571
Add and Manage Elements	2572
Add a Connector	2573
Add and Manage Diagrams	2574
Add and Delete Features	2575
Element Extras	2575
Repository Extras	2578
Stereotypes	2580
Work With Attributes	2581
Work With Methods	2582
Enterprise Architect Add-In Model	2585
Add-In Tasks	2586
Create Add-Ins	2587
Define Menu Items	2588
Deploy Add-Ins	2589
Tricks and Traps	2591
The Add-In Manager	2593
Add-In Search	2594
XML Format (Search Data)	2595
Add-In Events	2596
EA_Connect	2597
EA_Disconnect	2598
EA_GetMenuItems	2598
EA_GetMenuState	2599
EA_MenuClick	2601
EA_OnOutputItemClicked	2602
EA_OnOutputItemDoubleClicked	2603
EA_Show Help	2604
Broadcast Events	2604
Add-In License Management Events	2606
EA_AddinLicenseValidate	2606
EA_AddinLicenseGetDescription	2607
Compartment Events	2608

EA_QueryAvailableCompartments.....	2608
EA_GetCompartmentData.....	2609
Context Item Events.....	2611
EA_OnContextItemChanged.....	2612
EA_OnContextItemDoubleClicked.....	2613
EA_OnNotifyContextItemModified.....	2614
EA_FileClose.....	2615
EA_FileNew.....	2615
EA_FileOpen.....	2616
EA_OnPostCloseDiagram.....	2617
EA_OnPostInitialized.....	2617
EA_OnPostOpenDiagram.....	2618
EA_OnPostTransform.....	2619
EA_OnPreExitInstance.....	2619
EA_OnRetrieveModelTemplate.....	2620
Model Validation Broadcasts.....	2621
EA_OnInitializeUserRules.....	2622
EA_OnStartValidation.....	2623
EA_OnEndValidation.....	2623
EA_OnRunElementRule.....	2624
EA_OnRunPackageRule.....	2625
EA_OnRunDiagramRule.....	2625
EA_OnRunConnectorRule.....	2626
EA_OnRunAttributeRule.....	2627
EA_OnRunMethodRule.....	2628
EA_OnRunParameterRule.....	2629
Model Validation Example.....	2630
Post-New Events.....	2634
EA_OnPostNew Element.....	2635
EA_OnPostNew Connector.....	2636
EA_OnPostNew Diagram.....	2637
EA_OnPostNew DiagramObject.....	2638
EA_OnPostNew Attribute.....	2639
EA_OnPostNew Method.....	2640
EA_OnPostNew Package.....	2641
EA_OnPostNew GlossaryTerm.....	2642
Pre-Deletion Events.....	2642
EA_OnPreDeleteElement.....	2643
EA_OnPreDeleteAttribute.....	2644
EA_OnPreDeleteMethod.....	2645
EA_OnPreDeleteConnector.....	2646
EA_OnPreDeleteDiagram.....	2647
EA_OnPreDeleteDiagramObject.....	2647
EA_OnPreDeletePackage.....	2648
EA_OnPreDeleteGlossaryTerm.....	2649
Pre-New Events.....	2650
EA_OnPreNew Element.....	2651
EA_OnPreNew Connector.....	2652
EA_OnPreNew Diagram.....	2653
EA_OnPreNew DiagramObject.....	2654
EA_OnPreNew Attribute.....	2655
EA_OnPreNew Method.....	2656
EA_OnPreNew Package.....	2657
EA_OnPreNew GlossaryTerm.....	2658
Tagged Value Broadcasts.....	2658
EA_OnAttributeTagEdit.....	2659
EA_OnConnectorTagEdit.....	2660
EA_OnElementTagEdit.....	2661

EA_OnMethodTagEdit.....	2662
Technology Events.....	2663
EA_OnInitializeTechnologies.....	2663
EA_OnPreActivateTechnology.....	2664
EA_OnPostActivateTechnology.....	2665
EA_OnPreDeleteTechnology.....	2666
EA_OnDeleteTechnology.....	2667
EA_OnImportTechnology.....	2668
Custom Views	2669
Create a Custom View	2670
Custom Docked Window	2671
MDG Add-Ins	2671
MDG Events.....	2672
MDGBuild Project.....	2673
MDGConnect.....	2674
MDGDisconnect.....	2675
MDGGetConnectedPackages.....	2676
MDGGetProperty.....	2676
MDGMerge.....	2678
MDGNew Class.....	2680
MDGPostGenerate.....	2681
MDGPostMerge.....	2682
MDGPreGenerate.....	2683
MDGPreMerge.....	2683
MDGPreReverse.....	2684
MDGRunExe.....	2685
MDGView.....	2686

Part XX Glossary 2689

A	2690
B	2693
C	2695
D	2699
E	2702
F	2704
G	2706
H	2707
I	2708
J	2710
L	2711
M	2712
N	2715
O	2716
P	2717
Q	2721
R	2722
S	2725
T	2730
U	2732
V	2734

Part XXI License Management 2736

- Finding Your License Information 2738
- Adding License Keys 2739
- Keystore Troubleshooting 2742
- Upgrade an Existing License 2743
- Register Add-In 2745

Index 2747

Foreword

This user guide provides an introduction to the features contained in Enterprise Architect - a UML modeling tool for developing and building software systems with UML.

Part



1 Introduction



Sparx Systems **Enterprise Architect** is a full life-cycle **UML** based tool for:

- Designing and constructing a wide range of software systems
- Business analysis, business process modeling, managing requirements
- Systems modeling, systems architecture modeling, component design, simulation
- Generalized domain specific modeling
- Building domain specific modeling languages based on UML
- Visualizing a wide range of systems, processes, data, activities and structures
- Simulating behavioral processes, state machines and interactions
- Collaborating and sharing information and models
- Testing, quality control and end-to-end verification/traceability of complex systems
- Managing developmental tasks, work and scheduling

For over a decade, Sparx Systems has actively and continually developed Enterprise Architect based on the current UML standards. Over those years it has proven itself in a wide range of projects from small single user models to extremely large team based enterprise repositories. With over a quarter of million licensed users in over 100 countries world wide, the power and effectiveness of Enterprise Architect has been convincingly demonstrated in many different areas

Starting Out

Where to Start	Link
A summary of what Enterprise Architect can do and what you can use it for.	Overview ⁵ What You Can Do ⁷ Key Benefits ⁹ Key Features ¹⁴
A quick review of the process of modeling with Enterprise Architect. How to start Enterprise Architect and create a model project.	Getting Started ⁴⁴ A Quicksart Tutorial ⁴⁹
The initial tutorial has many links to more extensive descriptions of features, functions, tasks and procedures.	User Interface Guide ⁸⁶

Where to Start	Link
The Sparx Systems Formal Statements includes Copyright Notices, Acknowledgements and the End User License Agreement.	Formal Statements ^[27]
A glossary of terms is included that you can refer to for definitions of various terms and concepts.	Glossary ^[2689]
How to give feedback We value any suggestions, feedback and comments you might have regarding this product, documentation or install process. You can provide your feedback: <ul style="list-style-type: none">• Using a fault (bug) report or feature request, or• By email	www.sparxsystems.com/bug_report.htm www.sparxsystems.com/feature_request.htm support@sparxsystems.com
How to purchase Enterprise Architect and find pricing information.	Order Enterprise Architect ^[35]

1.1 Overview

Enterprise Architect

- Is a comprehensive **UML** analysis and design tool.
- Covers all aspects of **software**, **business** and **systems** modeling and design
- Is suitable for **requirements gathering** through analysis, design, construction, build, debug, simulation, testing, change management and maintenance to implementation, with full **traceability**.
- Combines the power of the latest UML specification (www.omg.org) with a high performance, intuitive interface, to bring an integrated and advanced toolset to the whole development team.
- Is a **scalable**, multi-user, visual tool with a rich feature set
- Helps build and document robust, maintainable systems and processes.
- Is a **proven**, highly popular tool for analysts, developers, consultants and managers in over 130 countries.
- Is used in the development of many kinds of application and system in a wide range of industries
- Covers **multiple domains**, including: aerospace, banking, web development, engineering, finance, medicine, military, research, academia, transport, retail, utilities and electrical engineering. It also has a long history of being used by standards organization worldwide to structure and organize their domain specific knowledge and communication channels.
- Is **supported** by many **partners**, **consultants**, colleges and other organizations who provide training and consulting services based around Enterprise Architect.

How this Help is Organized

Topic	Description	Link
Getting Started	Some introductory tutorials and important information to get up and running quickly.	Getting Started ^[44]
Setting up and managing Projects	Configuring, managing and maintaining modeling projects . I ncludes storage mechanisms, team development environments, managing changes and monitoring progress.	Projects and Teams ^[176]
Models and Modeling	Includes how to work with the models and components, their properties, and the reference data with which you populate the model.	Modeling Basics ^[657]
UML Concepts and Language	Basic UML and open standards based extensions. How those concepts are extended in Enterprise Architect's support of other modeling languages, and how you can create your own modeling languages.	Define a Modeling Language ^[1292]
Navigating and Searching	How to find the information you require in the model. Selecting aspects of the model to extract and review.	Navigate, Search & Trace ^[566]

Topic	Description	Link
	Tracing developmental relationships in both the structure and the development timeframe.	
Transforming	model elements and fragments from one domain to another, using standard and customized Model Driven Architecture transformations.	Model Transformation ^[1679]
Generating Code	Forward engineering from models and importing code (reverse engineering). Synchronizing code and model. Database engineering is also supported, including forward engineering and import of legacy database structures.	Software Engineering ^[1733] Database Engineering ^[1931] SOA and XML ^[2026]
Visual Analysis of Executing code	Advanced debugging and run-time inspection in a variety of languages. Tools for recording execution and stack traces and automatically generating Sequence diagrams. Profiling tools for native Windows applications. A powerful object workbench to test and work with individual objects. A rich testing capability featuring custom Testpoints based on Programming by Contract ideas and using a "code free" xUnit like approach.	Visual Execution Analyzer ^[2143] Profiling Native Applications ^[2170] Object Workbench ^[2176] Testpoint Management ^[2187] Unit Testing ^[2183]
Simulations	Run model Simulations on behavioral diagrams. Extensive support for State machines, Activity graphs, Interactions and BPMN. A powerful and easy to use feature that brings your model to life and helps ensure the validity and behavior of your solution.	Model Simulation ^[2100]
Domain Specific Modeling	Developing models for specific modeling domains .	Analysis and Business Modeling ^[1512] Database Engineering ^[1931] Systems Engineering ^[1984] SOA and XML Engineering ^[2026]
Quality Control	Ensuring quality control of your models and code through model validation and running your own test scripts .	Testing ^[2202]
Maintenance	Managing the maintenance of model elements and changes and issues across the project.	Maintenance ^[2233]

Topic	Description	Link
Reporting	Documenting your model in either RTF or HTML output format.	Reporting ^[2249]
Automation and Scripting	How to use the ActiveX automation interface provided by Enterprise Architect to script and extend the capabilities of the tool. How to write Add-ins and how to use the scripting capabilities.	Automation ^[2371] Scripting ^[2372] Enterprise Architect Add-In Model ^[2585]
Licensing	Managing your private or shared Enterprise Architect product license keys .	License Management ^[2736]

1.1.1 What You Can Do

Enterprise Architect is a rich and diverse modeling environment. At the highest level it uses a wide range of tools to help manage, understand, visualize, explore, trace and work with complex systems.

- General analysis, design, construction, visualization, testing, simulation and management of many domains
- Software construction, analysis, management, testing, coding, debugging and visualizing
- Business analysis, process definition, requirements management, process simulation, traceability
- Systems analysis, design, simulation, engineering and verification

A summary of some of the main activities supported by Enterprise Architect

Topic	Detail	See also
Modeling Analysis and Design Requirements	To create models with Enterprise Architect, you should become familiar with: <ul style="list-style-type: none"> • The UML standards and • How you apply UML in Enterprise Architect to develop your models • UML extensions such as SysML, BPMN, DDS and others • Domain based modeling • Model Transformations and MDA 	Modeling Fundamentals ^[657] Standard UML Models ^[1008] Requirement Models ^[1468] Domain Based Models ^[1498] Model Transformation ^[1679]
Managing Models	To manage the models in your projects you: <ul style="list-style-type: none"> • Create and structure models • Work with elements within packages and views 	Projects and Teams ^[178] Team Development ^[240] Reporting ^[2249]

	<ul style="list-style-type: none"> • Create diagrams as visual representation of model elements • Import and export model information • Protect and manage the model data itself, and • Communicate information on the data in the form of RTF and HTML documentation and reports • Managing change with baselines, XMI import and export and auditing 	Change Management [311]
Code Engineering	<p>In Enterprise Architect, UML modeling both depends on and supports code engineering - you generate and update code from a model, and you create and update models from code</p> <p>In this broad sense, Enterprise Architect enables you to:</p> <ul style="list-style-type: none"> • Forward engineer, reverse engineer, round-trip and synchronize code in a range of programming languages • Debug and profile code • Model and generate code for XML Technologies • Perform database modeling and design for a range of database management systems • Convert model components from one domain to another using Model Driven Architecture (MDA) Transformations 	Overview Of Development [1736]
Simulation		
Managing Projects	<p>Enterprise Architect provides strong support for Project Management, particularly in the following areas:</p> <ul style="list-style-type: none"> • Project estimation - working out how much time and effort is required to build and deploy a solution, using the Use Case metrics facility and carefully-calibrated metrics • Defining, assigning and managing resources • Communicating Project Management discussions and decisions to the project • Monitoring and managing problems, changes, issues and tasks that affect both individual elements and the project as a whole • Managing the development, execution and results of testing, from Integration through to User Acceptance, and • Maintaining a project glossary of terms, procedures and 	Projects and Teams [176] Testing [2202] Maintenance [2233] License Management [2736]

	<p>policies applied to the project</p> <p>The scope of your project management might include upgrades to Enterprise Architect and installation of related technologies</p>	
Extending Enterprise Architect Facilities	<p>Experienced technology developers can develop customized additions to the functionality already present within Enterprise Architect. These additions include:</p> <ul style="list-style-type: none"> • UML Profiles and Stereotypes • UML Patterns • Code Templates • Tagged Value Types • MDG Technologies and • Enterprise Architect Add-Ins. • Scripts <p>By creating these extensions the Technology Developer can customize the Enterprise Architect modeling process to specific tasks and speed up development.</p>	<p>Build Your Own Modeling Language ^[1292]</p> <p>Automation ^[2371]</p>

1.1.2 Key Benefits

Enterprise Architect is a powerful tool for specifying, documenting and building your software and business process projects.

Using Enterprise Architect's **support for UML** and its related standards, you can model new complex software and business systems, or visualize and maintain existing systems.

Key Benefits and Usage

Topic	Detail	See also
Design and Build Diverse Systems Using UML	<ul style="list-style-type: none"> • UML 2.3 is an open standard that provides a rich language for describing, documenting and designing software, business and IT systems in general. • Enterprise Architect enables you to leverage the full expressive power of UML 2.3 to model, design and build diverse systems in an open and well understood manner. • You can generate code, database structures, documentation and metrics; transform models; or specify behavior and structure as the basis for contractual agreements. 	<p>Standard UML Models ^[1008]</p> <p>Modeling Basics ^[657]</p>
Model and Manage	<ul style="list-style-type: none"> • Enterprise Architect helps individuals, groups and large 	<p>Configure User Security ^[251]</p>

Topic	Detail	See also
Complexity	<p>organizations model and manage complex information.</p> <ul style="list-style-type: none"> • Often this relates to software development and IT systems design and deployment, but it can also relate to business analysis and business process modeling. • Enterprise Architect integrates and connects a wide range of structural and behavioral information, helping to build a coherent and verifiable architectural model, either what-is or what-will-be. • Tools to manage version control, track and compare differences, audit changes and enforce security, help control project development and enforce compliance to standards. 	Change Management ^[31]
Structured Use Case Scenarios	<ul style="list-style-type: none"> • Enterprise Architect's Structured Scenario editor enables you to develop structured Use Case Scenarios, to capture vital analysis information in the form of natural language descriptions. • The editor helps you use this information to drive downstream development and maximize traceability across the development life-cycle. • The editor also helps you to dynamically link scenario steps to associated model elements, such as domain elements, business rules and glossary terms. • From structured scenarios, you can automatically generate test case descriptions, and Activity and other UML behavioral diagrams. • You can even reverse engineer existing process diagrams into structured, textual specifications to produce documentation deliverables. 	Scenarios ^[852]
Share Models	<ul style="list-style-type: none"> • Enterprise Architect enables you to share complete models or specific aspects of a model between members of a team, including (through the 'Lite', read-only edition) stakeholders who can study a model but not change or manage it. • You can make your project .EAP file available on a shared network drive, or replicate the .EAP file for complex distributed development. • Alternatively, you can develop the project in one of several shared DBMS repositories. 	The Read-only 'Lite' Edition ^[26] Project Sharing ^[240] Server Based Repositories ^[193] XML Import and Export ^[41] Version Control ^[312] Perform a Project

Topic	Detail	See also
	<ul style="list-style-type: none"> You can import and export data as XML files to distribute and update frameworks and other package-based model structures; you control changes through the version control repository. Enterprise Architect provides a data transfer wizard that enables you to upsize or downsize the complete model for maximum flexibility, and it enables you to export and import reference data so that you do not have to recreate it for related projects. 	Data Transfer ^[442] Sharing Reference Data ^[305]
Model, Manage and Trace Requirements	<ul style="list-style-type: none"> Enterprise Architect enables you to capture requirements and use full traceability from base requirements to design, build, deployment and beyond. You can use impact analysis to trace from proposed changes to original requirements, and build the 'right' system. 	Requirements ^[1468] Trace: Tracking Dependencies ^[635]
Develop Personal Views and Extracts of the Model	<ul style="list-style-type: none"> Enterprise Architect enables you to develop any number of different views of your model, or parts of it, either for your personal use or for the use of your team. These Model Views are generated by reports, so they can be set up to always show the current status of the selected view. The facility also enables you to create Favorites folders of hyperlinks to frequently-used data structures. 	Model Views ^[598]
Track and Trace Model Structures	<ul style="list-style-type: none"> In even a small model, it can be difficult to locate specific packages, diagrams or elements, even if you apply a rigorous naming and structure policy. Enterprise Architect has a wealth of facilities that enable you to locate structures quickly and easily, through the Model Search, Package Browser, Diagram List, Auditing facility, Traceability window, Relationship Matrix and reports. The Element menu, Diagram menu and Project Browser context menus also enable you to locate elements in diagrams and in the Project Browser, and you can store hyperlinks to important or commonly-used elements and diagrams in the Model Views. 	Navigate, Search & Trace ^[566] Auditing ^[382] Other Documents ^[2339] Main Menu ^[90] Project Browser Context Menus ^[569] Insert Related Elements ^[833]

Topic	Detail	See also
	<ul style="list-style-type: none"> Finally, having located one element you can import any related elements into a diagram in a single operation. 	
Generate Documentation	<ul style="list-style-type: none"> Enterprise Architect provides powerful document generation and reporting tools with a full WYSIWYG template editor for RTF or HTML output. You can generate complex and detailed reports from Enterprise Architect with the information you require in the format your company or client demands. 	Reporting ^[2249]
Generate and Reverse Engineer Source Code	<ul style="list-style-type: none"> Enterprise Architect supports generation and reverse engineering of source code for many popular languages. With a built in 'syntax highlighting' source code editor, Enterprise Architect enables you to quickly navigate and explore your model source code in the same environment. Code generation templates enable you to customize the generated source code to your company specifications. 	Generate Source Code ^[1768] Importing Source Code ^[1790] Editing Source Code ^[1799] Code and Transform Templates ^[1395]
Visualize, Inspect and Understand Complex Software	<ul style="list-style-type: none"> Software is complex and often hard to understand; you can use Enterprise Architect to reverse engineer code in a wide range of software development languages and database repository schema, to understand static structure. To complete the picture, use the unique built-in profiling and debugging tools to capture and visualize executing software at run-time. Create run-time instances of model elements and invoke methods using the built in Object Workbench. You can also bring in complete frameworks from source code or Java .jar files - or even .Net binary assemblies! By importing frameworks and library code, you can maximize re-use and understanding of your existing investment. 	Importing Source Code ^[1790] Execution Analysis ^[2143] Object Workbench ^[2176] Import Source Code ^[1792] Import Binary Module ^[1797]
Use Model Driven Architecture and Transformations	<ul style="list-style-type: none"> Model Driven Architecture (MDA) is an open standard designed to facilitate rapid application development in a platform independent manner. Models can be built at a high level of abstraction and, using MDA based tools, transformed into models and code 	Model Transformation ^[1679] Built-in Transformations ^[1686]

Topic	Detail	See also
	<p>targeting a specific platform or domain.</p> <ul style="list-style-type: none"> Enterprise Architect supports advanced MDA transformations using easily edited and developed transformation templates. With built-in transformations for DDL, C#, Java, EJB and XSD, you can quickly develop complex solutions from simple platform independent models (PIMs) targeted at platform specific models (PSMs); one PIM can be used to generate and synchronize multiple PSMs, providing a significant productivity boost. 	
SOA (Service Oriented Architecture) Support	<p>Enterprise Architect enables you to rapidly model and forward- and reverse-engineer two key W3C XML technologies:</p> <ul style="list-style-type: none"> XML Schema (XSD) and Web Service Definition Language (WSDL) <p>XSD and WSDL support is critical for the development of a complete Service Oriented Architecture (SOA), and the coupling of UML 2.3 and XML provides the natural mechanism for specifying, constructing and deploying XML-based SOA artifacts within an organization.</p>	<p>XML Schema - XSD ^[2027]</p> <p>Web Services - WSDL ^[2067]</p>
Systems Engineering support	<p>Integrating many high-end features for Systems Engineers, the Ultimate and Systems Engineering editions of Enterprise Architect provide built-in support for:</p> <ul style="list-style-type: none"> SysML Parametric model simulation Executable code generation Model to code transformations for Hardware Description Languages and Ada 2005 	<p>Editions Available ^[22]</p> <p>SysML ^[1990]</p> <p>Behavior ^[899]</p> <p>State Machine Modeling For HDLs ^[1784]</p> <p>Ada 2005 ^[1743]</p>
Model Databases	<ul style="list-style-type: none"> Enterprise Architect enables you to reverse engineer from many popular DBMS systems. You can model database tables, columns, keys, foreign keys and complex relationships using UML and an inbuilt data modeling profile, and forward generate DDL scripts to create target database structures. 	<p>Supported Databases ^[1970]</p> <p>Data Models ^[1625]</p>
Customize Enterprise	<ul style="list-style-type: none"> Enterprise Architect also includes facilities that enable 	<p>Developing Profiles ^[1293]</p>

Topic	Detail	See also
Architect	<p>experienced tool developers to customize and extend Enterprise Architect to suit the specific requirements of their organization with, for example, in-house UML Profiles, Add-Ins and Code Templates.</p> <ul style="list-style-type: none"> The very detailed Automation Interface gives you access to most element features, major functions such as XMI import/export, and attached information; most properties are fully writable from the automation client. The Automation Interface provides great support for plug-ins, with the ability to embed automation client windows in the main diagram view. The Interface is accessible from any automation-aware client language, such as VB, C#, C++ and Delphi. 	<p>Enterprise Architect Add-In Model^[2585]</p> <p>Code Template Framework^[1394]</p> <p>Using the Automation Interface^[2382]</p>
Link Enterprise Architect to IDEs	<ul style="list-style-type: none"> Using Sparx Systems Model Driven Generation (MDG) Link plug-ins, you can develop source code in your preferred Integrated Development Environment such as Visual Studio .NET or Eclipse, while you use Enterprise Architect to locate the source code for Classes, attributes and operations, and to model, navigate, track, reverse engineer, build and run your project. The MDG Integration products for Eclipse and Visual Studio 2008 provide an even closer, seamless integration of Enterprise Architect and UML 2.3 with your IDE, bringing the functionality required of a fully fledged modeling platform right inside the IDE. 	<p>Visual Studio .NET Eclipse (Link) Eclipse (Integration) Visual Studio 2008</p>

1.1.3 Key Features

Enterprise Architect has a rich and advanced set of tools for conducting a wide range of modeling, analysis, requirements, design, construction and visualization tasks. The table below lists some of the more important capabilities and how they assist you in your modeling and design tasks.

Key Features

Feature	Use To
Modeling	<ul style="list-style-type: none"> Model information, software, business rules and processes, hardware systems and more using UML compliant notation (latest UML specification - see www.omg.org)

Feature	Use To
	<ul style="list-style-type: none"> • Leverage SysML, BPMN, DDS and other domain specific UML extensions • Model dependencies between elements, system dynamics and state • Model class hierarchies, deployment, components and implementation details • Save and load complete diagrams as UML Patterns • Create and share dynamic views of model elements and diagram sets using Model Views • Analyze and trace relationships between elements using the tabular Relationship Matrix • Built-in Requirements Management enables you to specify, trace and verify requirements directly against the design, right through to the deployed solution • Traceability. Auditing. Impact analysis
Projects Sharing Import/Export	<ul style="list-style-type: none"> • Simple file based repositories for small teams • Scalable large team support using shared repositories (MS SQL Server, MySQL, Oracle and others supported) • Record project issues, tasks and system glossary • Assign resources to model elements and track effort expended against required effort • Manage personal tasks, work flow and collaborative model based mail • Schedule tasks and track important events using built in calendars and Gantt charts • Distributed development through shareable files, use of shared repositories in a range of major Database Management Systems, file replication, data transfer, and import and export of reference data • Share models using the latest XMI 2.1 format • Import models in XMI format from other tools (where compliant) • Audit all changes and report on recent activity • Visual and textual differencing based on package level baselines
Business	<ul style="list-style-type: none"> • Model and simulate with BPMN • Use structured scenarios to rapidly define use case based behavior • Import plain text use case descriptions into structured scenarios • Generate diagrams from structured scenarios • Generate BPEL scripts automatically from Business Process models • Business rules editor/designer • Generate executable business logic from rule tasks and trace to natural

Feature	Use To
	language business rules
Database Modeling	<ul style="list-style-type: none"> • Database modeling profile • Reverse engineer from a range of DBMSs via ODBC • Forward engineer DDL scripts to create database structures • Import database schema, XSD and WSDL source • ArcGIS support
Software	<ul style="list-style-type: none"> • Support for forward and reverse code engineering for many languages including: <ul style="list-style-type: none"> • <i>ActionScript 3.0,</i> • <i>Java,</i> • <i>C#,</i> • <i>C++,</i> • <i>VB.Net,</i> • <i>Delphi,</i> • <i>Visual Basic ,</i> • <i>Python,</i> • <i>PHP,</i> • Import .NET and Java binaries
Systems	<ul style="list-style-type: none"> • Simulate SysML parametric models • Simulate state and activity diagrams • Verilog, • VHDL • SystemC
Visual Execution Analyzer Debuggers Build Testing	<p>The Visual Execution Analyzer contains many important tools for building, verifying and testing your software. Some of the standout features include:</p> <ul style="list-style-type: none"> • Debug numerous languages, including C++, Java, C#, VB.net, PHP and others • Integrate coding and development with the UML model • Quickly search through source files and through your model during coding • Intellisense based on model data • Record and generate sequence diagrams from execution traces • Use profiling tools for native windows applications written in C/C++



Feature	Use To
	<ul style="list-style-type: none"> • Use the Object Workbench to test individual objects • Testpoint facility to quickly apply xUnit like tests to source without writing code • Generate Testpoint diagrams and sets from execution traces
Simulation	<p>Simulation of behavioral models provides numerous benefits, including:</p> <ul style="list-style-type: none"> • Verify the correct behavior of modeled state machines and activity graphs • Demonstrate modeled solutions to customers with attractive, configurable simulation walk-throughs • Simulate buttons presses and real-world events with triggers • Simulate rules and conditional execution with both model elements and Javascript based code • Record sets of triggers and automate playback based on scenarios • Walk through sequence diagrams
Domain Specific modeling and Frameworks	<p>Modeling languages and frameworks for specialized domains are easy to build and use within Enterprise Architect. Several are supplied by Sparx Systems, and some additional ones by 3rd parties. The tools available in Enterprise Architect also let you build your own frameworks for your special domain.</p> <ul style="list-style-type: none"> • Leverage industry-standard Enterprise Architecture frameworks such as Zachman, TOGAF, UPDM and others • Extended modeling for Requirements, User Interface Design, Mind Mapping, Data Modeling • SysML, DDS • BPMN • Built-in Model Driven Architecture (MDA) Transformations • Create custom transforms • Use UML Profiles to for domain-specific modeling • A range of add-Ins to integrate and extend the facilities of Enterprise Architect • Write your own add-ins • Transform behavioral models into executable source code for software and hardware description languages (HDLs) such as Verilog, VHDL, and SystemC
Reporting	<ul style="list-style-type: none"> • Comprehensive and flexible MS Word-compatible HTML and RTF report options • Read-only Viewer enables stakeholders to view but not change milestone deliverables

Feature	Use To
Manage Change	<ul style="list-style-type: none"> • Manage, track and control change using baseline model merge and auditing capabilities • Migrate changes across a distributed environment using Controlled XMI Packages • Manage Version control through XMI using SCC CVS and Subversion configurations • Inbuilt user and group security and access control management • Auditing
Testing	<ul style="list-style-type: none"> • Testing support for test cases, JUnit and NUnit • Integrated Debug Workbench for visualizing executable Java and .Net applications, instantiating run-time model objects and generating Sequence diagrams from a stack traces • Extensive Testpoint facility.
Automation and Scripting	<ul style="list-style-type: none"> • Script and automate common tasks using a detailed Automation Interface and Model Scripts
User Experience	<ul style="list-style-type: none"> • Speed: Enterprise Architect is quick to load and a spectacularly fast performer, even with large models • Scalability: Enterprise Architect supports single users and the development of small models, or many concurrent users developing extremely large models, with equal ease • Usability: many of our users agree, Enterprise Architect gets you started and productive quickly, with a rich user interface and the ability to create templates, model views and 'favorites' collections of commonly-used elements and diagrams.

For a complete list of the new features of the latest version of Enterprise Architect, click on the **Help | Read Me** menu option.

Enterprise Architect is available in six editions: **Ultimate, Business and Software Engineering, Systems Engineering, Corporate, Professional**, and **Desktop**, each of which offers a different range of features.

[Learn more](#)

- [Editions Available](#) 
- [Extensions - MDG Technologies](#) 

1.2 Enterprise Architect Editions



Enterprise Architect is available in a number of different editions, each tailored to support a particular business case

Trial and Beyond

Topic	Detail	See also
Try before Purchase	Test the product in a number of configurations in the Trial Version	The Trial Version ^[20]
Different Editions	Explore the specific features of the six work environment editions available	Editions Available ^[22]
The Lite Edition	Use the free, read-only or 'Lite' edition	The Read-only 'Lite' Edition ^[26]

1.2.1 The Trial Version

The fully functional 30 day trial version of Enterprise Architect is available free of charge on the Sparx Systems website www.sparxsystems.com/bin/easetup.exe. The trial version is identical to the registered edition with the exception that all diagrams are output to files with an embedded watermark.

Trial Details

Topic	Detail	See also
Try Out Editions	<p>The trial version enables you to explore and evaluate Enterprise Architect, checking the edition you are interested in and trying out other editions for comparison.</p> <p>When you start up the Enterprise Architect trial version, the Select Trial Version prompt displays; select the mode to trial.</p> <p>If necessary, you can close down Enterprise Architect and restart it in another mode for comparison.</p> <p>The prompt also directs you to useful information such as a walkthrough of the Enterprise Architect facilities, and enables you to use one of several workspace layouts.</p>	Workspace Layouts ^[149]
Trial Period:	As you are evaluating the Enterprise Architect trial version, note	Order Enterprise

Topic	Detail	See also
	<p>that the software operates for a limited period, and denies access after the trial period has elapsed.</p> <p>To continue using Enterprise Architect when the trial period expires, you can either:</p> <ul style="list-style-type: none"> • Apply to extend the trial period, or • Purchase and register a full license; on purchase of a suitable license or licenses, the registered version is made available for download <p>The latest information on pricing and purchasing is available on the Sparx Systems website; for more information, contact Sparx Systems by email.</p> <p>When you order and pay for an edition of Enterprise Architect, you receive installation instructions and the location of the executable files for download.</p>	<p>Architect^[35]</p> <p>Installation^[36]</p> <p>Register a Full License^[38]</p> <p>sales@sparxsystems.com</p>
Extend Trial Period	<p>If you are testing the trial version and require more than 30 days to evaluate it, you can apply to Sparx Systems Sales for an extension of the trial period; Sparx Systems Sales send you an extension key by email.</p> <p>The extension can be an additional 30 to 150 days; if you do not request a long enough extension in the first place, you can submit further requests.</p> <p>The trial period must expire before you can enter the extension key.</p>	<p>sales@sparxsystems.com</p>

How to

To extend the trial period, after receipt of your extension key


Step	Action	See also
1	<p>Open the Enterprise Architect trial version</p> <p>The Evaluation Version of Enterprise Architect dialog displays; once the trial period has expired, you cannot proceed beyond this dialog without extending the trial period</p>	
2	<p>Press (Ctrl) whilst you click on the Continue Trial button</p> <p>The Upgrade Key dialog displays</p>	
3	<p>In the Upgrade Key field, type or copy-and-paste the extension key you received from Sparx Systems Sales</p>	


Step	Action	See also
4	<p>Click on the OK button</p> <p>Enterprise Architect confirms that your trial period has been extended</p> <p>Your trial period is extended by the period of days stated in the email from Sparx Systems Sales; you can now restart and use the Trial version of Enterprise Architect</p>	


1.2.2 Editions Available



Enterprise Architect is available in six editions: the **Ultimate**, **Business and Software Engineering**, and **Systems Engineering** 'suite' editions, and the **Corporate**, **Professional** and **Desktop** editions.


Edition Comparison

Topic	Detail	See also
Comparison	<p>The functionality for each edition is described below; the features of and differences between the editions are listed in the table provided on the Sparx Systems website.</p> <p>To help you understand the differences between these editions and the advantages and limitations of each, you can open and use the Trial version of Enterprise Architect to mimic any of the editions.</p>	<p>Sparx Systems website</p> <p>The Trial Version ^[20]</p>
<p>Ultimate Edition</p> 	<p>The Ultimate edition is designed for power users and those working across multiple domains, providing deep support for Business, for Software Engineering and for Systems Development seamlessly integrated into a single development environment.</p> <p>It enables you to drill down to the lowest levels of systems design and construction, with SysML and executable code generation for standard and hardware description languages.</p> <p>Business users can leverage BPEL, the Rules Composer and executable UML, in addition to all the advanced features of the other editions of Enterprise Architect.</p> <p>Software developers can integrate their Eclipse and Visual Studio projects with their UML models and leverage the advanced executable code generators to target different domains.</p> <p>The Ultimate edition enables end to end traceability throughout a global vision of your enterprise - unifying strategy, business process, interfaces, software, rules, data and fine grained systems; powerful tools, domain-specific technologies, frameworks, integration platforms and a consistent, scalable, and robust interface work in unison to help you deliver on the promise of Model Driven Development.</p>	<p>MDG Technologies ^[1289]</p>

Topic	Detail	See also
	<p>The Ultimate edition incorporates a number of MDG Technologies and Add-Ins; the Ultimate edition and MDG Technologies are all available in either Fixed License or Floating License form.</p> <p>The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently.</p> <p>The Ultimate edition provides:</p> <ul style="list-style-type: none"> • Executable Code Generation - support for generating functional source code for State Machines, Interactions and Activities in C, C++, C#, Java and VB.NET • Full round trip support for Hardware Description Languages (Verilog, VHDL and SystemC) including support for generating State Machine code • SysML Simulation Support - including support for simulating SysML 1.2 constraint models with results graphing capabilities • BPEL Generation – transform BPMN 1.1 and BPMN 2.0 Business process models down to BPEL 1.1 and BPEL 2.0 code respectively • Business Rules – trace from abstract business rules down to automatically generated behavioral code 	
<p>Business and Software Engineering Edition</p> 	<p>The Business and Software Engineering edition is aimed at software development professionals, business modelers, architects, requirements experts, project managers and others involved in the design and construction of quality software and business services.</p> <p>It combines powerful new features such as executable code generation from UML models, BPEL, advanced scripting and a multi-purpose Rules Composer targeting executable code from Business Domain models; it also bundles licenses for integration products and frameworks such as DoDAF-MODAF, TOGAF and Zachman, to provide advanced model-driven construction tools to tightly bind your code development in Eclipse or Visual Studio.</p> <p>The Business and Software Engineering edition incorporates a number of MDG Technologies and Add-Ins; the Business and Software Engineering edition and MDG Technologies are all available in either Fixed License or Floating License form.</p> <p>The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently.</p>	<p>MDG Technologies [1289]</p>

Topic	Detail	See also
	<p>The Business and Software Engineering edition provides:</p> <ul style="list-style-type: none"> • Generation of Behavioral Code from State, Sequence and Activity models, supporting standard programming languages such as Java and .NET • Advanced math functions within the scripting engine • BPEL Generation from BPMN 1.1 and BPMN 2.0 models - including validation and WSDL support • A Business Rules Composer that enables you to build Business Domain models and generate code to implement complex business rules in standard programming languages 	
<p>Systems Engineering Edition</p> 	<p>The Systems Engineering edition is designed for systems and software development professionals working on real-time, embedded and systems solutions.</p> <p>It combines new features such as executable code generation from UML models (including support for hardware languages such as Verilog and VHDL), Ada, SysML 1.2, executable SysML Parametric diagrams and advanced scripting.</p> <p>It also bundles licenses for DoDAF-MODAF, SysML, DDS, TcSE and integration products to provide powerful model-driven construction tools for the Systems Engineering domain to tightly bind your code development in Eclipse or Visual Studio with the UML/SysML models developed in Enterprise Architect.</p> <p>The Systems Engineering edition incorporates a number of MDG Technologies and Add-Ins; the Systems Engineering edition and MDG Technologies are all available in either Fixed License or Floating License form.</p> <p>The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently.</p> <p>This edition provides:</p> <ul style="list-style-type: none"> • Executable Code Generation - support for generating functional source code for State Machines, Interactions and Activities in C, C++, C#, Java and VB.NET • Full round trip support for Hardware Description Languages, including Verilog, VHDL and SystemC, with support for generating State Machine code • SysML Simulation Support - Includes support for simulating SysML 1.2 constraint models with results graphing capabilities 	<p>MDG Technologies 1289</p>

Topic	Detail	See also
<p>Corporate Edition</p> 	<p>Aimed at larger development teams, the Corporate edition enables you to connect to the following DBMS back ends as the shared repository:</p> <ul style="list-style-type: none"> • MySQL • SQL Server • PostgreSQL, • Sybase Adaptive Server Anywhere • Access 2007 and • Oracle 9i, 10g or 11g <p>This provides additional scalability and improved concurrency over the shared .EAP file approach to model sharing.</p> <p>User security, user logins, user groups and user level locking of elements, user/group based security (with locking at diagram and element levels) are also supported.</p> <p>Security comes in two modes: in the first mode, all elements are considered 'writeable' until explicitly locked by a user or group; in the second mode, all elements are considered locked until checked out with a user lock.</p> <p>The Corporate edition forms the base for the three extended editions described above; like those editions, it is available in either Fixed License or Floating License form.</p> <p>The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently.</p>	
<p>Professional Edition</p> 	<p>Aimed at work groups and developers, the Professional edition supports shared projects through replication and shared network files.</p> <p>This edition has an ActiveX interface for interrogating Enterprise Architect projects and extracting information in XML format.</p> <p>The Professional edition fully supports code import/export and synchronization of model elements with source code; it enables reverse engineering of SQL Server; MS Access 97, 2000 and 2003; and Oracle 9i, 10g or 11g databases.</p> <p>Support for MDG Technologies and MDG Link (sold separately) is included with the Professional version.</p> <p>The shared repository available in the Professional edition is restricted to the .EAP file format (JET database).</p>	
<p>Desktop Edition</p>	<p>The Desktop edition is targeted at single analysts and developers producing UML analysis and design models.</p> <p>It provides facilities for:</p> <ul style="list-style-type: none"> • UML modeling 	

Topic	Detail	See also
	<ul style="list-style-type: none"> • XML import/export • Document generation • Version control integration and • Profile/metamodel extensibility 	

1.2.3 The Read-only 'Lite' Edition

Enterprise Architect Lite is a **free**, *read-only* edition of Enterprise Architect that enables people such as project sponsors to review the project without making any changes.

The read-only Lite Version

Topic	Detail	See also
Further Access	<p>Users of Enterprise Architect Lite also have wider access to</p> <ul style="list-style-type: none"> • The Team Review, where readers can create and respond to posts, and link their comments to elements • The Source Code Editor, where readers can open and edit external source code files, debug code, and configure and run Analyzer scripts • The File menu, where readers can copy the project or create a shortcut to access it • The Relationship Matrix, where readers can export the matrix contents to a CSV file to be opened in a spreadsheet application • The Default Hours tab to review project metrics. <p>You can download the Enterprise Architect Lite edition (as the Enterprise Architect Viewer) from the Sparx Systems website.</p>	<p>Team Review Tools ^[279]</p> <p>Editing Source Code ^[1799]</p> <p>File Menu ^[92]</p> <p>Relationship Matrix Options ^[645]</p> <p>Default Hours ^[520]</p> <p>http://www.sparxsystems.com/products/ea/downloads.html.</p>
Other Read-Only Options	<p>You can also make your model available to others in a read-only format by:</p> <ul style="list-style-type: none"> • Generating an HTML report on the model, which can be published on the web with read-only access 	<p>HTML Reports ^[2352]</p>

1.3 Formal Statements



Please take the time to read the following legal statements concerning Sparx Systems Enterprise Architect:

Legal Notes

Topic	Link
Software Copyright Notice	Software Copyright Notice ^[27]
Enterprise Architect End User Licensing Agreement	Enterprise Architect End User Licensing Agreement ^[28]
Acknowledgement of Trademarks	Acknowledgement of Trademarks ^[31]

Spark Systems would also like to gratefully [acknowledge contributions](#) ^[34] to the development of Enterprise Architect.

1.3.1 Copyright Notice

Copyright © 1998 - 2011 Sparx Systems Pty. Ltd. All rights reserved.

The software contains proprietary information of Sparx Systems Pty Ltd. It is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. Please read the [license agreement](#) ^[28] for full details.

Due to continued product development, this information can change without notice. The information and intellectual property contained herein is confidential between Sparx Systems and the client and remains the exclusive property of Sparx Systems. If you find any problems in the documentation, please report them to us in writing. Sparx Systems does not warrant that this document is error-free. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of Sparx Systems. Licensed users are granted the right to print a single hardcopy of the user manual per licensed copy of the software, but may not sell, distribute or otherwise dispose of the hardcopy without written consent of Sparx Systems.

Sparx Systems Pty. Ltd.

7 Curtis St,
Creswick, Victoria 3363,
AUSTRALIA

Phone: +61 (3) 5345 1140
Fax: +61 (3) 5345 1104

Support Email: support@sparxsystems.com
Sales Email: sales@sparxsystems.com

Website: www.sparxsystems.com

Scintilla and SciTE

Copyright 1998-2003 by Neil Hodgson <neilh@scintilla.org> All Rights Reserved.

Permission to use and distribute this (Scintilla) software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

1.3.2 End User License Agreement

Enterprise Architect Modeling Tool Version 9.2

Desktop, Professional, Corporate, Business and Software Engineering, Systems Engineering & Ultimate Editions

Copyright (C) 1998-2011 Sparx Systems Pty Ltd. All Rights Reserved

IMPORTANT- READ CAREFULLY: This End User License Agreement ("EULA") is a legal agreement between YOU as Licensee and SPARX SYSTEMS ("SPARX") for the SOFTWARE PRODUCT identified above. By installing, copying, or otherwise using the SOFTWARE PRODUCT, YOU agree to be bound by the terms of this EULA. If YOU do not agree to the terms of this EULA, promptly return the unused SOFTWARE PRODUCT to the place of purchase for a full refund.

The copyright in the SOFTWARE PRODUCT and its documentation is owned by Sparx Systems Pty Ltd A.C.N 085 034 546. Subject to the terms of this EULA, YOU are granted a non-exclusive right for the duration of the EULA to use the SOFTWARE PRODUCT. YOU do not acquire ownership of copyright or other intellectual property rights in any part of the SOFTWARE PRODUCT by virtue of this EULA.

Your use of this software indicates your acceptance of this EULA and warranty.

DEFINITIONS

In this End User License Agreement, unless the contrary intention appears:

- "ACADEMIC EDITION" means an edition of the SOFTWARE PRODUCT purchased for educational purposes at an academic discount price.
- "EULA" means this End User License Agreement.
- "SPARX" means Sparx Systems Pty Ltd A.C.N 085 034 546.
- "Licensee" means YOU, or the organization (if any) on whose behalf YOU are taking the EULA.
- "Registered Edition of Enterprise Architect" means the edition of the SOFTWARE PRODUCT which is available for purchase from the web site: http://www.sparxsystems.com/ea_purchase.htm.

- "SOFTWARE PRODUCT" or "SOFTWARE" means Enterprise Architect, UML Case Tool, Desk Top, Professional, Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions, which includes computer software and associated media and printed materials, and may include online or electronic documentation.
- "Support Services" means email based support provided by SPARX, including advice on usage of Enterprise Architect, investigation of bugs, fixes, repairs of models if and when appropriate, and general product support.
- "SPARX support engineers" means employees of SPARX who provide on-line support services.
- "Trial edition of Enterprise Architect" means the edition of the SOFTWARE PRODUCT which is available free of charge for evaluation purposes for a period of thirty (30) days.
- "EALITE" means the LITE version of Enterprise Architect that is distributed free of charge as a read-only viewer of .EAP files.

GRANT OF LICENSE

In accordance with the terms of this EULA YOU are granted the following rights:

- a) To install and use one copy of the SOFTWARE PRODUCT or, in its place, any prior version for the same operating system, on a single computer. As the primary user of the computer on which the SOFTWARE PRODUCT is installed, YOU may make a second copy for your exclusive use on either a home or portable computer.
- b) To store or install a copy of the SOFTWARE PRODUCT on a storage device, such as a network server, used only to install or run the SOFTWARE PRODUCT over an internal network. If YOU want to increase the number of users entitled to concurrently access the SOFTWARE PRODUCT, YOU must notify SPARX and agree to pay an additional fee.
- c) To make copies of the SOFTWARE PRODUCT for backup and archival purposes only.

EVALUATION LICENSE

The Trial version of Enterprise Architect is not free software. Subject to the terms of this agreement, YOU are hereby licensed to use this software for evaluation purposes without charge for a period of thirty (30) days.

Upon expiration of the thirty (30) days, the SOFTWARE PRODUCT must be removed from the computer. Unregistered use of Enterprise Architect after the 30-day evaluation period is in violation of Australian, U.S. and international copyright laws.

SPARX may extend the evaluation period on request and at their discretion.

If YOU choose to use this software after the 30 day evaluation period a license must be purchased (as described at http://www.sparxsystems.com/ea_purchase.htm). Upon payment of the license fee, YOU will be sent details on where to download the registered edition of Enterprise Architect and will be provided with a suitable software 'key' by email.

EALITE

Subject to the terms of this Agreement [EALITE](#)^[26] may be installed on any machine indefinitely and free of charge. There are no fees or Sparx support services in relation to EALITE.

ADDITIONAL RIGHTS AND LIMITATIONS

YOU hereby undertake not to sell, rent, lease, translate, adapt, vary, modify, decompile, disassemble, reverse engineer, create derivative works of, modify, sub-license, loan or distribute the SOFTWARE PRODUCT other than as expressly authorized by this EULA.

YOU further undertake not to reproduce or distribute license key-codes except under the express and written permission of SPARX.

If the Software Product purchased is an Academic Edition, YOU ACKNOWLEDGE THAT the license is limited to use in an educational context, either for self-education or use in a registered teaching institution. The Academic Edition may not be used to produce commercial software products or be used in a commercial environment, without the express written permission of SPARX.

ASSIGNMENT

YOU may only assign all your rights and obligations under this EULA to another party if YOU supply to the transferee a copy of this EULA and all other documentation including proof of ownership. Your license is then terminated.

TERMINATION

Without prejudice to any other rights, SPARX may terminate this EULA if YOU fail to comply with the terms and conditions. Upon termination YOU or YOUR representative shall destroy all copies of the SOFTWARE PRODUCT and all of its component parts or otherwise return or dispose of such material in the manner directed by SPARX.

WARRANTIES AND LIABILITY

WARRANTIES

SPARX warrants that the SOFTWARE PRODUCT will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of receipt, and any Support Services provided by SPARX shall be substantially as described in applicable written materials provided to YOU by SPARX, and SPARX support engineers will make commercially reasonable efforts to solve any problems associated with the SOFTWARE PRODUCT.

EXCLUSIONS

To the maximum extent permitted by law, SPARX excludes, for itself and for any supplier of software incorporated in the SOFTWARE PRODUCT, all liability for all claims, expenses, losses, damages and costs made against or incurred or suffered by YOU directly or indirectly (including without limitation lost costs, profits and data) arising out of:

- YOUR use or misuse of the SOFTWARE PRODUCT
- YOUR inability to use or obtain access to the SOFTWARE PRODUCT
- Negligence of SPARX or its employees, contractors or agents, or of any supplier of software incorporated in the SOFTWARE PRODUCT, in connection with the performance of SPARX' obligations under this EULA, or
- Termination of this EULA by either party for any reason.

LIMITATION

The SOFTWARE PRODUCT and any documentation are provided "AS IS" and all warranties whether express, implied, statutory or otherwise, relating in any way to the subject matter of this EULA or to this EULA generally, including without limitation, warranties as to: quality, fitness; merchantability; correctness; accuracy; reliability; correspondence with any description or sample, meeting your or any other requirements; uninterrupted use; compliance with any relevant legislation and being error or virus free are excluded. Where any legislation implies in this EULA any term, and that legislation avoids or prohibits provisions in a contract excluding or modifying such a term, such term shall be deemed to be included in this EULA. However, the liability of SPARX for any breach of such term shall if permitted by legislation be limited, at SPARX's option to any one or more of the following upon return of the SOFTWARE PRODUCT and a copy of the receipt:

- If the breach relates to the SOFTWARE PRODUCT:
 - the replacement of the SOFTWARE PRODUCT or the supply of an equivalent SOFTWARE PRODUCT
 - the repair of such SOFTWARE PRODUCT
 - the payment of the cost of replacing the SOFTWARE PRODUCT or of acquiring an equivalent SOFTWARE PRODUCT, or
 - the payment of the cost of having the SOFTWARE PRODUCT repaired.
- If the breach relates to services in relation to the SOFTWARE PRODUCT:
 - the supplying of the services again, or
 - the payment of the cost of having the services supplied again.

TRADEMARKS

All names of products and companies used in this EULA, the SOFTWARE PRODUCT, or the enclosed documentation may be trademarks of their corresponding owners. Their use in this EULA is intended to be in compliance with the respective guidelines and licenses.

Windows®, Windows 98, Windows NT, Windows ME, Windows XP, Windows Vista, Windows 2000 and Windows 2003 Server are trademarks of Microsoft®.

GOVERNING LAW

This agreement shall be construed in accordance with the laws of the Commonwealth of AUSTRALIA, in the state of Victoria.

1.3.3 Trademarks

Sparx Systems acknowledge the following trademarks and registered trademarks, which are used throughout the Enterprise Architect documentation.

Trademarks

Topic	Trademarks	Registered Trademarks
Microsoft Corporation	ActiveX™ Microsoft Office™ Microsoft Word™	Access® Active Directory® Excel®

Topic	Trademarks	Registered Trademarks
	VB.NET™	IntelliSense® JScript® Microsoft® MS Access® MS Access 2007® MSDE® SQL Server® Visio® Visual Basic® Visual SourceSafe® Visual Studio® Win32® Windows® Windows Vista®
The Object Management Group (OMG)	OMG™ Object Management Group™ The CORBA logo ORB™ Object Request Broker™ The CORBA Academy design OMG Interface Definition Language™ IDL™ CORBAservices™ CORBAfacilities™ CORBAmed™ CORBAnet™ Unified Modeling Language™ MOF™ CWM™ BPMN™ SoaML™ Model Driven Architecture™ MDA™ OMG Model Driven Architecture™ OMG MDA™	The OMG Object Management Group logo The Information Brokerage® CORBA® CORBA Academy® IIOP® XMI® UML® The UML Cube logo

Topic	Trademarks	Registered Trademarks
The Open Group		Archimate®
Methodologies Corporation	SOMF™	
Embarcadero Technologies Inc	InterBase™	Delphi®
Oracle Corporation	MySQL™	Oracle® Java®
Python Software Foundation	Python™	
Esri	ArcGIS™ ArcCatalog™	
Linux Mark Institute	Linux™	
Innobase Oy	InnoDB™	
Firebird Foundation Incorporated		FireBird®
Ingres Corporation	Ingres™	
IBM	DB2™ Informix™	Rational® Rational® Rhapsody
Sybase Inc	Sybase Central™	Adaptive Server® Adaptive Server® Anywhere Adaptive Server® Everywhere SQL Anywhere® Sybase®

Topic	Trademarks	Registered Trademarks
Progress Software Corporation		OpenEdge® Progress OpenEdge®

1.3.4 Acknowledgements

Some parts of this application include code originally written by various authors and modified for use in Enterprise Architect.

Acknowledgements

Name	Detail	Link
Marquet Mike	Print listview contents	mike.marquet@altavista.net
Davide Pizzolato	CXImage Library © 7-Aug-2001	ing.davide.pizzolato@libero.it
Neil Hodgson	Scintilla editor © 1998-2003	neilh@scintilla.org
Others	Many thanks to all those who have made suggestions, reported bugs, offered feedback and helped with the beta-testing of Enterprise Architect Your help has been invaluable	

1.4 Order Enterprise Architect

Enterprise Architect is designed, built and published by Sparx Systems, and available through the Sparx Systems website

- Free, 30-day trial version (although you can extend the evaluation period)
- Registered purchase version that provides the facilities of the edition you require, as unlocked by the license keys that you purchase from Sparx Systems
- The trial version of Enterprise Architect is identical to the registered version with the exception that all diagrams are output to files with an embedded watermark. The trial software stops working after the trial period has elapsed.
- On purchase of a suitable license or licenses, the registered version is made available for download.
- The latest information on pricing and purchasing is available on the Purchase and Pricing page of the Sparx Systems website, and from Sparx Sales.

Purchase Options

- On-line using a secure credit-card transaction
- Fax
- Check or equivalent
- Bank transfer

Learn more

- [The Trial Version](#)^[20]
- [Editions Available](#)^[22]
- sales@sparxsystems.com
- [Pricing and Purchase Options](#)

1.5 Installation

Enterprise Architect is distributed as a single executable setup file (.exe). It is a standard Windows installer based file that will check and update previous installations of Enterprise Architect.

- After you install Enterprise Architect, you can immediately begin to create projects as .EAP files.
- The latest evaluation and registered versions of Enterprise Architect are always available from the Sparx Systems website.
- The registered version is available through the registered user area of the web site, which requires a username and password to access.
- These are provided upon purchase of a license.

Installation

Topic	Detail	See also
System Requirements	The system requirements for installing Enterprise Architect are defined on the Enterprise Architect System Requirements page of the Sparx Systems website.	Enterprise Architect System Requirements
Windows Vista / Windows 7	<p>Under Windows Vista and Windows 7 (with User Account Control turned on) an application starts with only Standard permissions, regardless of what level of authority the current user has.</p> <p>As a result, an installer run normally with an Admin account under Vista only has Standard privileges and either is not able to write to certain critical areas of the registry/file system, or redirects the write requests to a per-user virtualized registry/file system.</p> <p>Sparx Systems recommend that if you are installing on Windows Vista or Windows 7, always run the Enterprise Architect installer with Administrator privileges (right-click on the downloaded installer icon and select the Run as administrator menu option).</p>	
Install Enterprise Architect	<p>Run the Enterprise Architect setup program. Generally you can accept all the default options without change.</p> <p>To place Enterprise Architect in a directory other than the default, enter the name of the destination when prompted.</p> <p>You might be prompted to restart your computer when the installation completes; although this is not always necessary (if you already have the components Enterprise Architect requires installed on your computer), you should restart just to be certain.</p>	
Installation on Linux and MacOS	If you intend to run Enterprise Architect on Linux, refer to the page <i>Installing Enterprise Architect inside Wine</i> , on the Sparx Systems website.	Installation inside Wine

Topic	Detail	See also
	To support the Scripting facility in the Corporate and extended editions of Enterprise Architect, you must also install Internet Explorer 6.0 or later revisions.	
Using a Third Party DBMS as Model Repository	<p>If you plan to use SQL Server, MySQL, PostgreSQL, Access 2007, Sybase Adaptive Server Anywhere or Oracle 9i, 10g or 11g as a model repository, then you must use the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions of Enterprise Architect.</p> <p>You also require additional files and supplementary installation processes; please note that installation and maintenance of these database management systems is not covered under the support agreement.</p> <p>Users planning to use SQL Server, MySQL, PostgreSQL, Sybase Adaptive Server Anywhere, Access 2007 or Oracle 9i, 10g or 11g as their model repository can access scripts that create the required data structures for the choice of DBMS, from the Sparx Systems website; to access these scripts, please follow the links at right.</p>	<p>Corporate Edition Resources Page</p> <p>Trial Corporate Edition Resources Page</p>

Notes

- Enterprise Architect requires **Read/Write** access to the program files directory where Enterprise Architect has been installed

1.6 Register a Full License

The **Trial** version of Enterprise Architect available for download is an evaluation version only. For the **full** version you must first purchase one or more *licenses*.



The license key supplied determines which edition (Enterprise Architect Desktop, Professional, Corporate, Business and Software Engineering, System Engineering or Ultimate) is activated after installation.

The Corporate, Business and Software Engineering, System Engineering and Ultimate editions can be activated by either a private license key or a shared (floating) license key. Shared license keys, for a user population that includes temporary users, are checked out of and in to a Sparx Systems key store.

Use to

- Obtain and activate the registered version of Enterprise Architect

How to

Step	Action	See also
1	<p>Purchase one or more licenses.</p> <p>Once you have paid for a licensed version of Enterprise Architect, you receive (via email or other suitable means):</p> <ul style="list-style-type: none"> • a license key or keys • a web site address from which to download the full version • if you have purchased a Sparx Systems key store, a web site address from which to download the key store 	Editions Available 
2	Save the license key(s) and download the latest full install package from the address supplied.	
3	Run the setup program to install the full version.	
4	Open Enterprise Architect from the Start Menu or desktop icon. The License Management dialog displays.	
5	Click on the Add Key button. The Add Registration Key dialog displays; register your key.	Add License Key 
6	Click on the OK button. The full version is now activated on your PC, and Enterprise Architect displays the message: <i>Registration succeeded! Thank you for purchasing Enterprise Architect <type> Edition</i>	

Step	Action	See also

Notes

- Private license keys and shared license keys have different formats, so you cannot use one in place of the other

Learn more

- [Upgrade an Existing License](#)^[2743]
- [Register Add-In](#)^[2745]

1.7 Help and Support




Enterprise Architect has three main help and information systems to assist you in using the product:

- Learning Center
- Enterprise Architect Help
- The Sparx Systems website

In addition Sparx Systems recommend that you fully explore the sample project supplied with Enterprise Architect. It assists you in learning to use Enterprise Architect and offers tips on getting the most out of Enterprise Architect's features. Click on the **EAExample** option on the Enterprise Architect Start Page.

Getting Help and Support

Topic	Detail	See also
Learning Center	<p>The Enterprise Architect Learning Center provides context-sensitive guidance, tools, demonstrations and other online resources to help you understand any area of Enterprise Architect that you are interested in.</p> <p>The Learning Center automatically displays on the right of the screen when you first open Enterprise Architect, showing the Getting Started topics. You can select other task areas by clicking on the drop-down arrow in the toolbar.</p>	Learning Center [89]
Enterprise Architect Help	<p>Enterprise Architect Help provides comprehensive documentation of Enterprise Architect and covers every aspect and facility of the product. To access Help within Enterprise Architect:</p> <ul style="list-style-type: none"> • Click on the Help icon () in the various toolbars • Select the Help Help Contents menu option • Click on the Help button on a dialog (for Help specific to that dialog) <p>Enterprise Architect Help is extensive; if you cannot quickly locate the topic you require in the online contents list, you can use one of two search facilities:</p> <ul style="list-style-type: none"> • Click on the Index tab, type in a keyword or key phrase appropriate to the subject you require help for, and press (Enter); double-click on the appropriate index item • Click on the Search tab, type in a word or phrase to search for, and click on the List Topics button; double-click on the 	Available Help File Formats [41]

Topic	Detail	See also
	<p>required topic.</p> <p>The Enterprise Architect Help is also available separately from the product, in different formats.</p>	
Sparx Systems Web Site	<p>The Sparx Systems web site is also extensive, and provides information and announcements concerning the company and its full range of products, as well as tutorials, white papers, templates and solutions.</p> <p>It also provides a user forum, community site (see below) and support network; Sparx Systems are highly responsive to user feedback and requirements, and the web site enables rapid communication concerning problems, solutions and enhancements.</p> <p>You can access the web page and user forum within Enterprise Architect from the View More Project Tools Internal Web Browser menu option, and through the Online Resources topics in the Learning Center.</p>	<p>Sparx Systems Web Site</p> <p>Enterprise Architect User Forum</p> <p>Learning Center</p>
Community Site	<p>The Sparx Systems website also hosts the Sparx Systems Enterprise Architect Community Site. This is a central location for the Enterprise Architect community to publish resources and share experiences.</p> <p>From this site you can download the latest news, tutorials, resources, best practices, tips, techniques and user-generated content for Enterprise Architect. You can also, as a registered author, contribute content and share your expertise with the wider community.</p>	<p>Sparx Systems Enterprise Architect Community Site</p>

Learn more

- [Contacting Sparx Support](#)^[42]

1.7.1 Available Helpfile Formats

You can access the latest Enterprise Architect help files from the following locations

Format	Link
.CHM	www.sparxsystems.com/bin/EA.chm
.CHM inside a .ZIP file:	www.sparxsystems.com/bin/EAHelp.zip
.PDF	www.sparxsystems.com/bin/EAGuide.pdf

Format	Link
.HTML	www.sparxsystems.com/EAUserGuide/index.html

Version and release date information for the help files can be found at

Topic	Link
Help file	www.sparxsystems.com/ea_downloads.htm#Helpfiles
Help file(registered users).	www.sparxsystems.com/registered/reg_ea_down.htm#Helpfiles

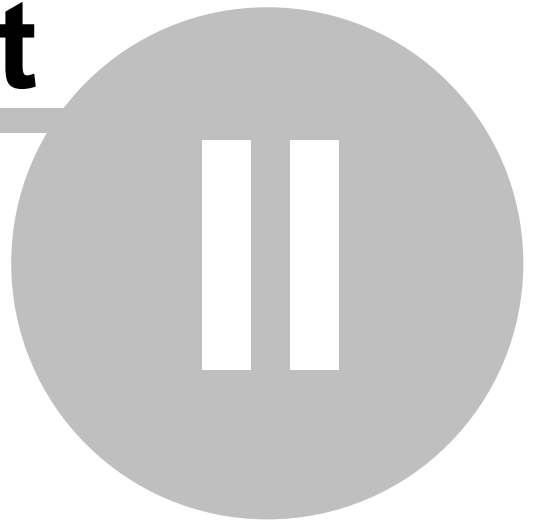
1.7.2 Support

Technical support for Enterprise Architect is available to registered users. Responses to support queries are sent by email.

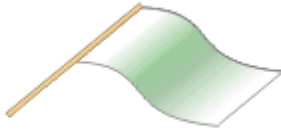
Sparx Systems endeavors to provide a rapid response to all product-related questions or concerns.

- Registered users can lodge a support request, by visiting: http://www.sparxsystems.com/registered/reg_support.html
- Trial users can contact Sparx Systems with questions regarding their evaluation at: support@sparxsystems.com
- An online user forum is also available for your questions and perusal, at <http://www.sparxsystems.com/cgi-bin/yabb/YaBB.cgi>

Part



2 Getting Started



This guide helps you to understand the options available when you start Enterprise Architect, and to quickly gain an understanding of how to use these options to create models.

Getting Started

Topic	Link
After starting Enterprise Architect, use the Quick Start tutorial to immediately create a project	A Quickstart Tutorial ^[49]
Read about work areas applicable to certain Project Roles	Project Roles ^[72]
Review the Enterprise Architect User Interface, or workspace	Enterprise Architect User Interface ^[86]

At various points throughout the Enterprise Architect Help, there are further Quick Start topics and sections to help you use the system immediately to experiment with a feature of Enterprise Architect. Use the Help Index tab and search for *Quick Start* to locate these topics.

Learn more

- [Common Tasks](#) ^[45]
- [Basics](#) ^[47]

2.1 Common Tasks

Enterprise Architect is a powerful modeling tool that can be used by project managers, business analysts, developers and testers alike to build business, real-time and software systems.

Common tasks and tools

Topic	Detail
Manage Projects Effectively	<p>Enterprise Architect provides all the tools necessary for the planning, execution and successful completion of a project.</p> <ul style="list-style-type: none"> • Create a shared vision of your project, improve communication and help team members agree upon the design of a proposed system • Evaluate project risk factors, report any changes or defects and develop strategies to tackle potential problems • Help estimate the amount of time and effort required to complete a project and maximize staff utilization • Use Auditing, Project Baselines and User Access Controls to better manage change in a collaborative environment
Analyze Strategic Business Needs	<p>A Business Analyst is responsible for gathering requirements, modeling potential solutions and evaluating the business needs of an organization.</p> <ul style="list-style-type: none"> • Build a coherent picture of how a business operates through requirements gathering, Use Case analysis and modeling business rules • Help achieve business process improvement and implement change using tools such as Strategic Models, Use Case Scenarios, Business Rule Models, Flow Charts, Auditing and the Team Review window
Build and Develop Systems	<p>The developer is responsible for implementing a solution using a range of tools.</p> <ul style="list-style-type: none"> • Bring the power and benefits of Enterprise Architect into your favorite Integrated Development Environment • Use Enterprise Architect's in-built support for writing code, including syntax highlighting, line numbering, auto completion, bookmarking and automatic indentation • Create breakpoints to step through code execution, examine variables and view error messages via the Debug window • Apply Visual Execution Analysis to identify costly function calls, explain system behavior and establish the sequence of events that occur immediately prior to system failure • Use Enterprise Architect's support for round trip engineering and

Topic	Detail
	<p>synchronizing source code with a corresponding model</p> <ul style="list-style-type: none">• In addition to many popular programming languages, create powerful database solutions that automatically generate DDL scripts for more than 10 different Database Management Systems
Test and Investigate	<p>Quality Assurance teams test software to identify defects, verify that it satisfies all requirements and ensure that it produces expected results.</p> <ul style="list-style-type: none">• In addition to the integrated JUnit and NUnit testing capabilities, Enterprise Architect enables you to create and manage test scripts for model elements, covering unit, integration, scenario, system and acceptance tests• Visualize the execution of code to better understand how applications work and the sequence of events that leads to program failure

2.2 Basics



When you install Enterprise Architect on your computer, a new program folder called *Enterprise Architect* is created in your **Start** menu (unless you changed the default name during installation).

You can start Enterprise Architect from the icon created on your Windows desktop during installation, or alternatively:

1. Open the Windows **Start** menu.
2. Locate the Enterprise Architect program folder.
3. Select **Enterprise Architect**.

After a short pause, the Start Page displays. From this page you can

Topic	Link
Open a project file (.EAP file)	Open a project file ^[18]
Create a new project (.EAP file)	Create a new project ^[18]
Connect to a DBMS repository (Corporate and extended editions)	Connect to a DBMS repository ^[20]

Notes

- By default, when you install Enterprise Architect, an empty 'starter' project called EABase.EAP is installed, as well as an example project named EAExample.EAP. We recommend that new users select the EAExample file and explore it in some detail while you become familiar with UML and software engineering using Enterprise Architect
- Enterprise Architect.EAP files default to use Jet 3.5 as the database engine. This does not support unicode character sets. If you want to use unicode character sets (for example, to provide user interface texts in languages other than your Windows-defined native language), you must either upsize to a DBMS repository or set JET 4.0 as the database engine. Resetting the database engine ensures compatibility with .EAP files that support unicode character sets and that are in turn compatible with versions of MS Access later than Access 97

If your .EAP project is not in a Jet 4.0 database, you should also download a copy of the Jet 4.0 EABase model from the Sparx Systems website, and do an EAP to EAP transfer of your model into the Jet 4.0

file

Learn more

- [Start Page](#)^[87]
- [A Quick Start Tutorial](#)^[49], to begin a guided exploration of Enterprise Architect immediately
- [Common Tasks](#)^[45], to get an overview of the kind of work you might perform with Enterprise Architect
- [Upsize to a DBMS Repository](#)^[193]
- [General Options](#)^[544] to set JET 4.0 as database engine
- [Sparx Systems website](#) for the Jet 4.0 EABase model
- [Perform a Data Transfer](#)^[442]

2.3 A Quickstart Tutorial



Welcome to Enterprise Architect!

This quick-start tutorial helps you get up to speed with Enterprise Architect.

As you read through this tutorial, it is recommended that you have Enterprise Architect open so that you can try out the tasks described. By the end you should be able to begin modeling your own business/software projects with Enterprise Architect.

Basic Tasks

Topic	Link
Creating a new project	Creating a New Project ^[49]
Adding Views to your model	Views ^[51]
Adding Packages to your model	Packages ^[52]
Adding diagrams	Adding Diagrams ^[52]
Adding elements to your model packages	Elements ^[53]
Creating links between model elements	Links ^[55]
Refining the model	Refining the Model ^[57]

Throughout the descriptions there are hyperlinks to more detailed information on a range of topics.

2.3.1 Create Project Scenario

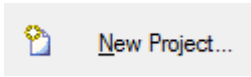
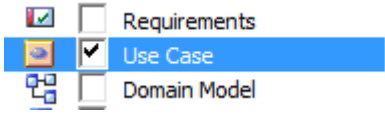
A project is a single file or repository-based store for one or more Models.

- The first step in getting started with Enterprise Architect is to either open an existing project, or create a new one
- In this example we create a new file-based project and add some template-based models to kick-start

our modeling

- Once your new project has been created it will have a simple Use Case View added for you to explore and tailor to your requirements
- You can re-open your project at any time by double-clicking on it in a file browser
- It should also appear in your Recent Projects list on the Start page

Create a New Project

Step	Action	Result
1	Start Enterprise Architect.	The Open Project dialog displays. (If the dialog does not appear, press (Ctrl+O) to invoke it.)
2	Click the New Project button to select a suitable location and name for your new project. 	The standard Windows file browser dialog displays. File-based Enterprise Architect projects are named with a .EAP extension.
3	Locate a suitable folder for your project and enter a distinctive name. Press the Save button to continue.	Enterprise Architect creates a new project file and places it in the specified location. The project is then opened automatically and the New Model Wizard displays.
4	In the left hand column ensure that Default is selected - and in the right hand side tick the Use Case View check box.  Click on the OK button.	The Model Wizard automatically creates a new Use Case model for you, with an initial diagram, some notes and default elements to help get you started. At this stage all your changes have been saved to file and do not require further action.


2.3.2 Add a View To Your Model

A View is a top level package, within a model. It can be categorized by purpose using different icon types, such as Use Case View, Component View or Deployment View. Views are used to contain packages, diagrams and elements - the building blocks of your model.

Use to

- Create a new View in your project

How to add a View

Step	Action	See also
1	<p>Select a model root node in the Project Browser, then either:</p> <ul style="list-style-type: none"> • Click the Add a Package button () on the Project Browser toolbar • Select Project New Package • Right-click the model root node and, from the context menu, select New View, or • Press (Ctrl + W) <p>The Create New View dialog displays.</p>	Views ⁶⁷⁶
2	Enter an appropriate name for the new View.	
3	Click the appropriate radio button to select an icon for the View.	
4	<p>Click on the OK button.</p> <p>The Create New View dialog closes.</p> <p>The new View is created as a child of the selected model root node.</p>	

Notes

- There are six types of View, which represent conventional ways of categorizing the purpose of a Model View; these Views represent different structural or behavioral aspects of the same model

Learn more


- [Add a package to your model](#) ⁵²¹
- [Create a new diagram](#) ⁵²¹
- [Add an element](#) ⁵³¹
- [Read more about views](#) ⁶⁷⁶

2.3.3 Add Packages

A **Package** is a container of model elements, and is displayed in the Project Browser as a 'folder' icon.

- A Package is required to hold the model building blocks, such as diagrams, elements and other packages
- To begin developing your Model, you must create an initial package for your first diagrams
- Note - when the **Automatically add new Diagram** option is selected, Enterprise Architect also creates a default diagram within the new package, to speed your development

How to add new Packages to your model


Step	Action	See also
1	<p>Select a Package or View in the Project Browser, then either:</p> <ul style="list-style-type: none"> • Click the Add a Package button  on the Project Browser toolbar. • Press (Ctrl + W) • Select Project New Package, or • Right-click the Package or View and, from the context menu, select Add Add Package <p>The New Model Package dialog displays.</p>	<p>Creating Diagrams ⁵²</p> <p>Add Elements ⁵³</p> <p>Move Components ⁵⁷</p> <p>Learn more about the Project Browser ⁵⁶</p>
2	Enter a suitable name for the package.	
3	Optionally, clear the check mark from Automatically add new diagram .	
4	<p>Click on the OK button.</p> <p>The New Model Package dialog closes.</p> <p>A new package is created as a child of the selected package or View.</p>	

2.3.4 Creating Diagrams

A diagram is a visual representation of

- The elements of your model
- Their attributes and characteristics
- How they are connected and/or interact with each other
- Different diagram types show different aspects of the model and the relationships between elements.

How to Add new diagrams to your model

Step	Action	See also
1	<p>Select a package or View in the Project Browser, then either:</p> <ul style="list-style-type: none"> Click on the Add a Diagram button  on the Project Browser toolbar Press (Ctrl + Insert) Select Project New Diagram, or Right-click the package or View and, from the context menu, select Add Add Diagram <p>The New Diagram dialog displays.</p>	<p>Add Elements ^[53]</p> <p>Move Elements ^[57]</p> <p>Learn more about UML diagrams ^[1010]</p>
2	<p>Enterprise Architect provides a default diagram name, that is the same name as the parent package.</p> <p>Update the diagram name as required.</p>	
3	<p>Select a category of diagrams in the left hand pane.</p> <p>The list of available diagram types in the right hand pane is updates according to your selection.</p>	
4	<p>Select the type of diagram to add from the right hand pane.</p>	
5	<p>Click on the OK button.</p> <p>The New Diagram dialog closes.</p> <p>A new diagram is created as a child of the currently selected package.</p> <p>The new diagram is opened in the Diagram View, ready for editing.</p>	

Notes

- When you create a package, if you leave the **Automatically add new Diagram** option selected, the New Diagram dialog displays automatically

2.3.5 Creating New Elements on Diagrams

Models are constructed from elements, each of which has its own meaning, rules and notation. Generally, when you create a new element in your model, you want to use that element on a diagram. The simplest way to do this is to create the element directly on the diagram.

How to Create new elements on diagrams in your model

Step	Action	See also
1	<p>Ensure that the Diagram Toolbox is visible.</p> <p>To display the Diagram Toolbox, either:</p> <ul style="list-style-type: none"> • Select View Diagram Toolbox, or • Press (Alt + 5) 	<p>Paste From the Project Browser^[733]</p> <p>Add connectors^[55]</p> <p>The Quick Linker^[79]</p> <p>Move elements^[57]</p> <p>UML elements^[1090]</p>
2	<p>Ensure that the diagram on which the elements are to be created is open in the Diagram View.</p> <p>To open the diagram, double-click on the diagram name in the Project Browser.</p> <p>The diagram opens and the Toolbox is updated to display the categories of elements and relationships that are applicable to that type of diagram.</p>	<p>Diagram View^[69]</p>
3	<p>Click on the appropriate icon in the Toolbox to select the type of element to create.</p> <p>The element type is highlighted in the Toolbox.</p>	
4	<p>Click on the diagram at the location where you want to place the element.</p> <p>The new element is created as a child of the package that contains the diagram, and is placed onto the diagram at the cursor position.</p> <p>The element Properties dialog displays.</p>	
5	<p>Use the Properties dialog to define characteristics of the element as required.</p>	<p>Properties Dialog^[843]</p>
6	<p>Click on the OK button.</p> <p>The Properties dialog closes.</p>	

Notes

- You can also drag or paste existing elements onto a diagram from the Project Browser
- If you are creating several elements of one type, after creating the first just press (**Shift+F3**) or (**Ctrl +Click**) to create the next element of that type

2.3.6 Adding Connectors

Connectors define specific relationships between specific elements, so you usually create them directly on the diagram by dragging the required relationship type from the Diagram Toolbox. As for elements, the Toolbox automatically presents the connector or relationship types appropriate to the type of diagram.

How to Define a relationship between two model elements on a diagram

Step	Action	See also
1	<p>Ensure that the Diagram Toolbox is visible.</p> <p>To display the Diagram Toolbox, either:</p> <ul style="list-style-type: none"> • Select View Diagram Toolbox or • Press (Alt + 5) 	
2	<p>Ensure that the diagram containing the elements to be connected, is open in the Diagram View.</p> <p>To open the diagram, double-click on the diagram name in the Project Browser.</p> <p>The selected diagram opens and the Toolbox is updated to display the categories of elements and relationships that are applicable to that diagram type.</p>	Diagram View ^[69] Creating Elements on Diagrams ^[53]
3	<p>Click on the required connector in the Toolbox to select the type of connector to draw.</p> <p>The connector type is highlighted in the Toolbox.</p>	
4	<p>Click on the source element in the relationship, then drag across to the target element.</p> <p>The selected connector is drawn between the two elements.</p> <p>The connector Properties dialog displays.</p>	
5	<p>Use the Properties dialog to define the characteristics of the connector as required.</p>	Properties Dialog ^[843]
6	<p>Click on the OK button.</p> <p>The Properties dialog closes.</p>	

Notes

- If you are creating several connectors of one type, after creating the first just click on the appropriate source element and press (**F3**) to create the next connector of that type

- As you drag a connector, you can press (**Shift**) to create a bend in the connector; if necessary, you can put several bends in the connector line, pressing (**Shift**) every time you want to change direction
- To roll back the bends, keep holding the left mouse button down and press (**Backspace**) as many times as is necessary
- To find out more about the type of connector you have dragged on to a diagram, right-click on the connector and select the **UML Help** menu option, which displays a Help page on the connector type

Learn more

- [Quick Linker - Create Connectors](#)^[793]
- [Create Connector in Project Browser](#)^[952]
- [UML Connections](#)^[1208]

2.3.7 Modify Properties

When you create an element and connect it to another element, you usually have to define various characteristics of both the element and the connector to identify the purpose and function they represent. You do this using a Properties dialog.

When you create elements, Enterprise Architect automatically names and numbers them by type - for example, Class1, Class2 - so you should at least change the **Name** field to more easily identify each element. Enterprise Architect does not automatically name connectors, but for many connector types you should provide a name that describes the purpose of the connection.

How to modify the characteristics of a model element or connector

Step	Action	See also
1	<p>Either:</p> <ul style="list-style-type: none"> • Double-click on an element or connector in the diagram • Right-click an element in the Project Browser and, from the context menu, select Properties, or • Right-click a connector in a diagram and, from the context menu, select Properties <p>The appropriate Properties dialog displays.</p>	<p>Element Properties Dialog^[843]</p> <p>Connector Properties Dialog^[961]</p>
2	Modify the element or connector characteristics as required.	<p>Element Properties Dialog^[843]</p> <p>Connector Properties Dialog^[961]</p>
3	<p>Click on the OK button.</p> <p>The Properties dialog closes.</p> <p>The modifications are saved; the Project Browser and diagrams that contain the element are updated as necessary.</p>	

Step	Action	See also

Notes

- Enterprise Architect is initially configured to display the Properties dialog automatically when you create an element or connector, but it is easy (and often convenient) to turn the dialog display off
- If the default display has been turned off, you can display the Properties dialog by either double-clicking on the element or connector in the diagram or by right-clicking on it in the Project Browser and selecting the **Properties** menu option

2.3.8 Moving Elements, Diagrams and Connectors

You have created a project containing packages, diagrams and elements, and you have connected the elements.

How do you change where things are?

How to move diagrams, elements and connectors

Topic	Link
Changing the Project Browser list order - re-arrange the order of items in the Project Browser	Move Elements Within a Package ^[58]
Moving elements between packages - move diagram, elements and child packages, from one parent package to another	Move Elements Between Packages ^[58]
Moving elements on a diagram - re-arrange the position of diagram elements	Move Elements in a Diagram ^[59]
Moving elements between diagrams - cut an element from one diagram and paste it onto another	Move Elements Between Diagrams ^[59]
Moving connectors on a diagram - re-link a source element to a different target element	Move Connectors on a Diagram ^[62]

Learn more



- [Delete Components](#) ^[62]
- [Arrange Connectors](#) ^[94]
- [Order Package Contents](#) ^[58]

2.3.8.1 Move Elements Within a Package

In the Project Browser, the contents of a package are listed in the order: diagrams, child packages, elements.

- Elements are further arranged in order of type
- Within a type, components are initially listed in alphabetical or numerical order
- You can change the order in which items are listed within their groups.

How to Adjust the order in which items are listed in the Project Browser

Step	Action	See also
1	Click on an item in the Project Browser, then click on  or  in the toolbar at the top of the window. The selected item is moved up or down in the Project Browser accordingly, but remains within its particular group of items.	
2	To revert to listing components in alphabetical order, in the Project Browser right-click on the package and from the context menu select Contents Reset Sort Order .	

2.3.8.2 Move Elements Between Packages

When creating the various diagrams, elements and packages that make up your model, at some point you might create an item under the wrong parent package.

In Enterprise Architect it is quite simple to relocate a model item from one parent package to another. Items can be relocated to a higher level package, a lower level package or a different package at the same level.

Use to

- Move a diagram from one package to another
- Move an element from one package to another
- Move a package from one parent package to another

How to move elements between packages

Step	Action	See also
1	In the Project Browser, click and drag the model item to be relocated from its existing position, and drop it onto the new parent package. The dragged item and all of its child items are relocated to the target package.	

Notes

- Moving elements in the Project Browser does not affect the use of elements in diagrams
- Moving an element or package has no effect on any relationships that the element, package, or elements within the package have
- Moving a diagram generally does not affect the location of elements in packages
- However, elements of certain types, such as Initial Node, Decision and Final Node, are used only within one diagram and have no meaning outside that diagram; if you move a diagram containing these elements, they are moved to the new parent package with the diagram

Learn more

- [Initial Node](#) ^[1136]
- [Decision](#) ^[1117]
- [Final Node](#) ^[1127]

2.3.8.3 Move Elements in a Diagram

To improve the understanding of a diagram or to simply improve the presentation, you might want to adjust the position of the elements on your diagram.

How to Adjust the position of the elements on your diagram

For coarse adjustments

Step	Action	See also
1	Click on a diagram element and drag it to its new position.	

For fine adjustments

Step	Action	See also
1	Click on a diagram element to select it.	
2	While holding down the (Shift) key, press the arrow keys to move the selected element one pixel at a time in the direction of the arrow (Shift + ↑, ↓, →, ←).	



2.3.8.4 Move Elements Between Diagrams

If an element is in the wrong diagram, Enterprise Architect offers functionality that simplifies the process of moving the element from one diagram to another.



Use to

- Move elements from one diagram to another
- Copy elements from one diagram to another

How to move elements from one diagram to another

Step	Action	See also
1	In the Project Browser, double-click the name of the diagram that contains the element to be moved. The diagram opens in the Diagram View.	
2	Click on the element in the diagram to select it.	
3	Either: <ul style="list-style-type: none"> • Click on the Cut button () in the Default Tools toolbar, or • Press (Ctrl + X) <p>The selected element is copied to the clipboard.</p> <p>However, the element remains visible on the current diagram until the clipboard content is pasted onto a different diagram.</p> <p>The model element itself is not affected in any way.</p>	
4	In the Project Browser, double-click on the name of the diagram into which the element is to be moved. The diagram opens in the Diagram View.	
5	Either: <ul style="list-style-type: none"> • Click on the Paste button () in the Default Tools toolbar, or • Press (Ctrl + V) or (Shift + Insert) <p>The element is pasted from the clipboard onto the current diagram and removed from the original (source) diagram.</p> <p>Again, the model element itself is not affected in any way.</p>	

How to copy elements from one diagram to another

Step	Action	See also
1	In the Project Browser, double-click the name of the diagram that contains the element to be moved. The diagram opens in the Diagram View.	
2	Click on the element in the diagram to select it.	
3	Either: <ul style="list-style-type: none"> Click on the Copy button () in the Default Tools toolbar, or Press (Ctrl + C) <p>The selected element is copied to the clipboard. Neither the current diagram, nor the model element itself are affected in any way.</p>	
4	In the Project Browser, double-click on the name of the diagram into which the element is to be copied. The diagram opens in the Diagram View.	
5	Either: <ul style="list-style-type: none"> Click on the Paste button () in the Default Tools toolbar, or Press (Ctrl + V) or (Shift + Insert) <p>The element is pasted from the clipboard onto the current diagram. Again, neither the source diagram nor the model element itself are affected in any way.</p>	

Notes

- By selecting more than one element at a time, you can move or copy multiple elements in the same operation
- You can remove an element from a diagram by selecting it, then pressing the Delete key
- You can place an element onto an open diagram by locating the element in the Project Browser, then dragging it onto the diagram


2.3.8.5 Connect a Different Element on a Diagram

When creating connectors in your diagrams, at some point you might connect the wrong pair of elements. Enterprise Architect provides a simple method to enable you to move the end of a connector from one element to another.

Use to

- Change the source element of a connector from the current element to another
- Change the target element of a connector from the current element to another

How to change a connected element

Step	Action	See also
1	In the diagram, click on the connector to select it. The connector is highlighted with 'handles' at each connector end.	
2	Move the cursor over the 'handle' at the end of the connector to reconnect. The cursor changes to a small arrow () pointing upwards to the right.	
3	Click and drag the connector end to the other element to connect to. When you release the mouse button, the connector breaks from the original element and reconnects to the new element.	

Notes

- You can also tidy up a connection by dragging the end of the connector to a better position on the edge of the element, or move both ends at once by dragging the middle of the connector

2.3.9 Delete Elements

You can delete the elements of a model from a diagram or from the Project Browser.

How to delete an element

Topic	Link
Remove elements from a diagram: this is essentially 'hiding' the element in that particular view of the model	Remove Elements from a Diagram ^[63]
Delete elements from a model - the element, its properties and any child elements or diagrams it has are all deleted; the element is removed from all diagrams on which it appears	Delete Elements from your Model ^[64]

Topic	Link

Learn more

- [Delete Connectors](#) ^[65]


2.3.9.1 Remove Elements From a Diagram

Removing an element from a diagram does not delete the element from the model and therefore does not alter the structure of the underlying model. When you remove an element from a diagram, you are essentially 'hiding' the element in that particular view of the model.

How to remove elements from a diagram

Step	Action	See also
1	<p>Open the diagram that is to be modified.</p> <p>Either:</p> <ul style="list-style-type: none"> • Double-click on the diagram name in the Project Browser, or • Make an already open diagram the 'active' diagram, by clicking on its tab in the Diagram View 	
2	<p>Either:</p> <ul style="list-style-type: none"> • Press (Ctrl+A) to select all of the elements in the diagram, or • Click on an element in the diagram to select it (and use (Ctrl + Click) to add further elements to your selection, or to remove elements from the selection) <p>The selected elements are highlighted in the diagram.</p>	
3	<p>Either:</p> <ul style="list-style-type: none"> • Press (Delete), or • Press (Ctrl+D), or • Right-click a selected element and from the context menu select Delete <element name> or Delete Selected Elements <p>The selected elements are removed from the diagram.</p>	

Notes

- You can 'undo' the removal of a diagram element, by pressing (**Ctrl + Z**), or by clicking the **Undo** button , in the Default Tools toolbar

2.3.9.2 Delete Elements From Your Model

You can delete elements from your model, working either from the Project Browser or from a diagram.

- When you delete an element from your model, the element, its properties and any child elements or diagrams it contains are all deleted.
- The deleted element is removed from all diagrams on which it appears
- All relationships that it was involved in are also deleted from the model.

How to delete one or more items from your model using the Project Browser

Step	Action	See also
1	In the Project Browser, click on an item to select it. To select additional items, or to remove items from the selection, use (Ctrl + Click) or (Shift + Click). The selected items are highlighted in the Project Browser.	
2	Either: <ul style="list-style-type: none"> Press (Ctrl + Delete) or Right-click on one of the selected elements, then select the Delete selected item(s) context menu option A confirmation dialog is displayed.	
3	Click on the Yes button. The selected items are deleted from your model.	

How to delete one or more items from your model using a diagram

Step	Action	See also
1	To open the diagram containing the items to be deleted from your model, either: <ul style="list-style-type: none"> Double-click on the diagram name in the Project Browser, or Make an already open diagram the 'active' diagram, by clicking on its tab in the Diagram View 	

Step	Action	See also
2	<p>Either:</p> <ul style="list-style-type: none"> • Press (Ctrl+A) to select all of the elements in the diagram, or • Click on an element in the diagram to select it, then use (Ctrl + Click) to add or remove specific elements to and from the selection <p>The selected elements are highlighted in the diagram.</p>	
3	<p>Press (Ctrl+Delete).</p> <p>A confirmation dialog is displayed.</p>	
4	<p>Click on the Yes button.</p> <p>The selected elements are removed from the model.</p>	

Notes

- Deleting a package completely removes the package and all its contents - diagrams, child packages and elements - from the model
- Deleting an element completely removes the element and its properties, connectors, child elements and child diagrams from the model, and from every diagram that contains it
- Deleting a diagram completely removes the diagram from the model, but not the diagram's component elements; they remain in the parent package

2.3.10 Delete Connectors

It is possible to hide a connector on a diagram, or to delete it entirely from the model, removing the relationship between the previously connected elements.

Displaying all of the connectors on a complex diagram can make the diagram somewhat confused, so it can be useful to hide some of the connectors, to clarify a specific aspect of a more complex picture.

Use to

- Hide a connector on a diagram
- Delete a relationship between two elements in your model

How to hide or remove a connector between two elements

Step	Action	See also
1	<p>Open the diagram containing the connector of interest.</p> <p>Either:</p>	

Step	Action	See also
	<ul style="list-style-type: none"> • Double-click the diagram name in the Project Browser, or • Make an already open diagram the 'active' diagram, by clicking on its tab in the Diagram View 	
2	<p>Click on the connector in the diagram to select it.</p> <p>The selected connector is highlighted in the diagram.</p>	
3	<p>Either:</p> <ul style="list-style-type: none"> • Press (Ctrl + Delete) • Right-click on the connector and, from the context menu, select Delete Connector • Press (Delete) <p>If you press (Ctrl + Delete), a dialog displays prompting you to confirm that you want to delete the connector (and the relationship it describes) from the model.</p> <p>Otherwise, the Remove Connector dialog displays.</p>	
4	<p>On the Remove Connector dialog, choose one of the options:</p> <ul style="list-style-type: none"> • Hide the connector • Delete the connector from the model <p>Hide the connector obscures the connector on the current diagram; you cannot see it but the relationship between the two elements still exists and is shown on other diagrams containing the two elements together, and on all reports of connectors between the elements.</p> <p>Delete the Connector from Model removes the relationship that exists between the two elements; the connector is removed from the current diagram, from all other diagrams on which it is shown, and from all reports on connectors between the two elements.</p>	<p>Hiding the Connector^[955]</p> <p>Relationships^[648]</p>
5	<p>Optionally, tick the checkbox Don't ask again.</p> <p>Selecting this option prevents the Remove Connector dialog from being displayed next time you select the Delete Connector command; the command uses the setting you last used on the dialog.</p> <p>Make sure that you have selected the right option to use as your default.</p>	
6	<p>Click on the OK button.</p> <p>The connector disappears from the diagram.</p> <p>If you selected Delete the connector from the model, the relationship represented by the connector is also deleted from your model.</p>	

Notes

- You cannot select multiple connectors - it is only possible to select one connector at a time
- If you select the **Don't ask again** checkbox on the Remove Connector dialog, you can reset this option on the Links page of the Options dialog (**Tools | Options | Links**)
- Selecting the **Hide the connector** option in the Remove Connector dialog, has the same effect as hiding the connector on the Links tab of the source element Properties dialog, or using the **Visibility | Hide Connector** context menu option

Learn more


- To identify and reveal hidden connectors, see the [Links](#) ^[85] topic

2.3.11 Save Changes

Throughout much of your work in Enterprise Architect, any changes you make are automatically saved when you close the *dialog* (data entry window) on which you made the changes. In some cases the dialog contains a **Save** or **Apply** button, which enables you to save your

changes and then keep working on the dialog.

If there is no specific dialog, such as when you create a diagram, you can save your work by

- Clicking on the **Save** icon in the Diagram toolbar ()
- Pressing the (**Ctrl + S**) keyboard keys, or
- Selecting the **Diagram | Save** menu option

Often, Enterprise Architect does not let you close a screen without confirming that you want to save or discard your changes. You can also save your diagram changes automatically, by selecting the **Auto Save Changes** checkbox on the Diagram Behavior page of the Options dialog.

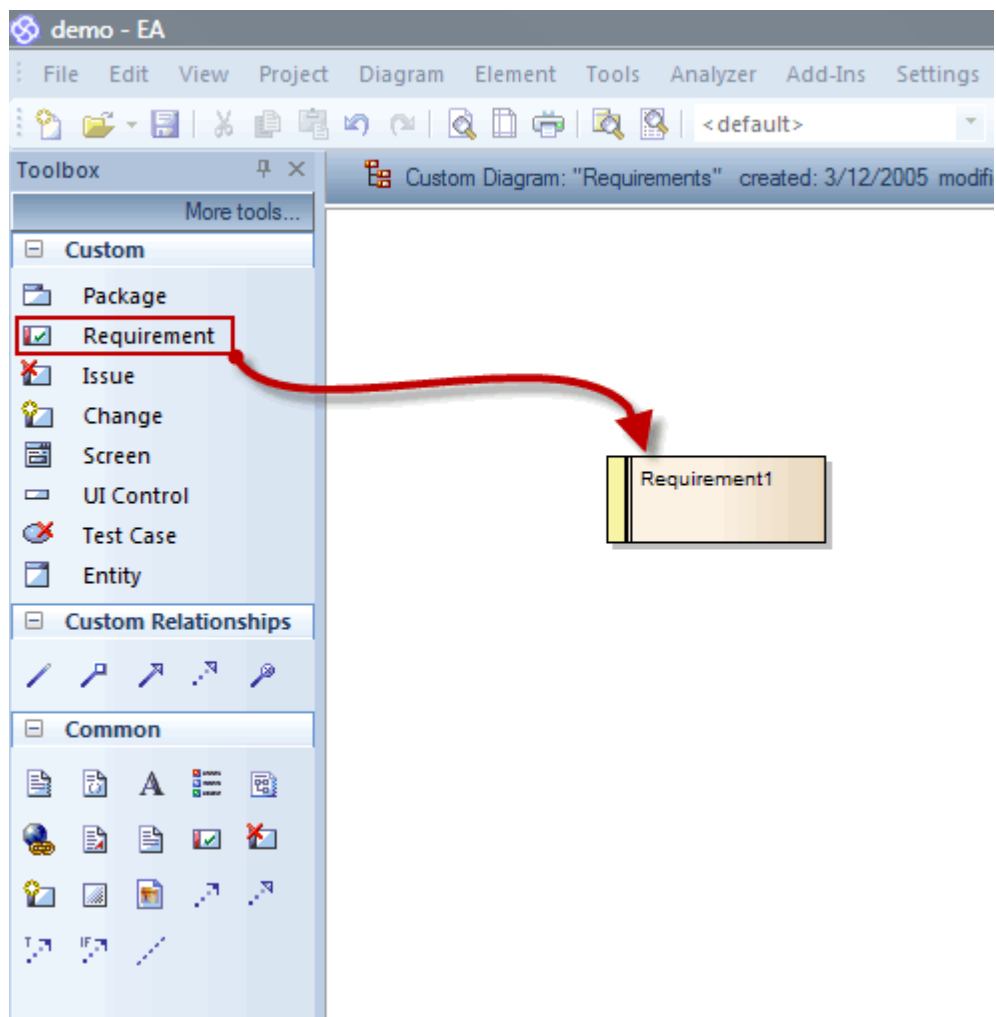
Learn more

- So far you have been using the Project Browser and Diagram View to develop your project; at this point you should find out a bit more about the other facilities of the Enterprise Architect [User Interface](#) ^[86] or Application Workspace
- When you have finished exploring the User Interface topics, go to the [Summary of Typical Tasks](#) ^[72] to identify areas of Enterprise Architect that provide particular support for your job role
- Explore other options for managing the way your diagrams display and function, on the [Diagram Behavior](#) ^[53] page

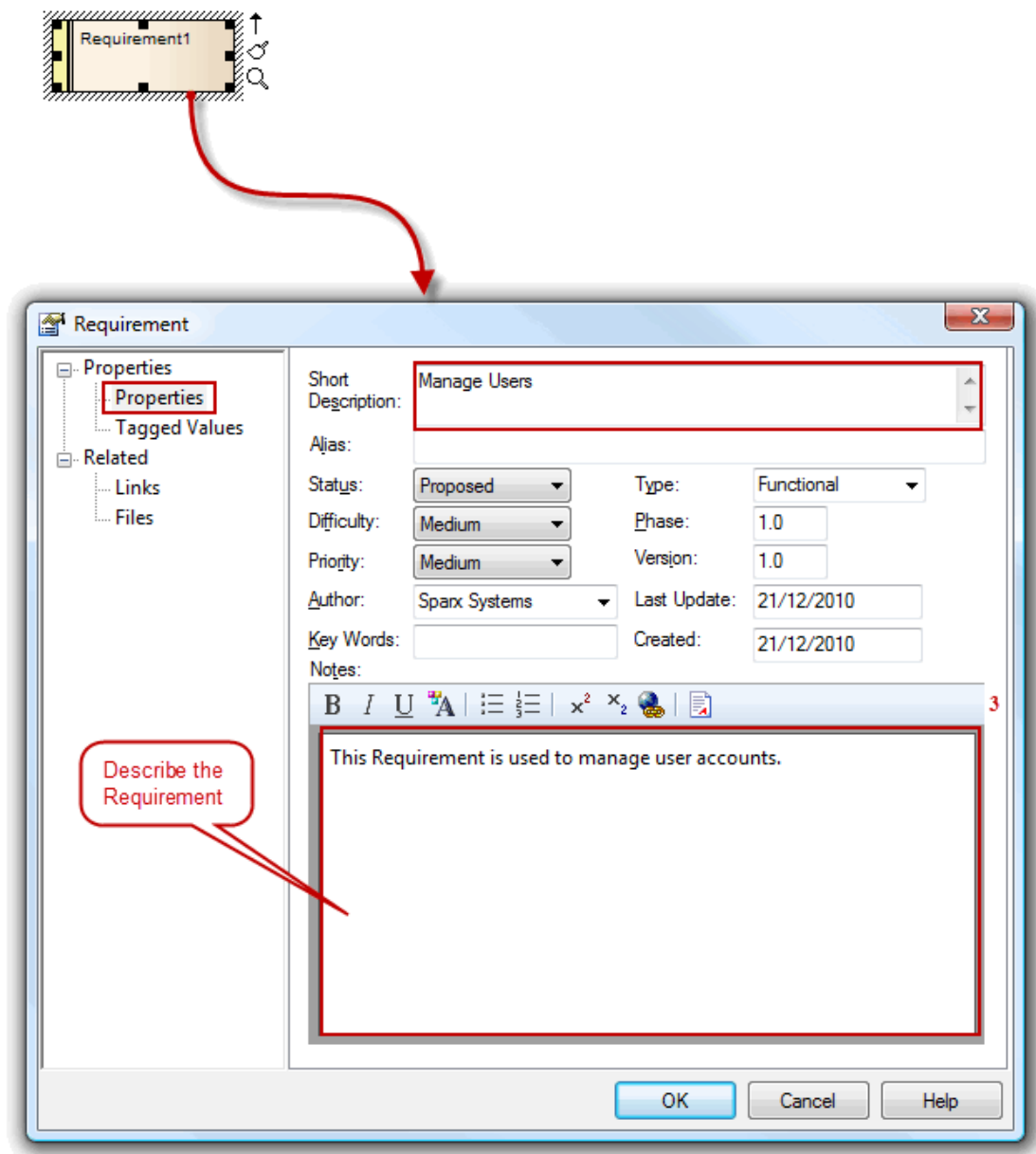
2.3.12 Create Requirements

The following example illustrates how to create a requirement, name that element and provide appropriate notes.

How to create a requirement, name that element and provide appropriate notes

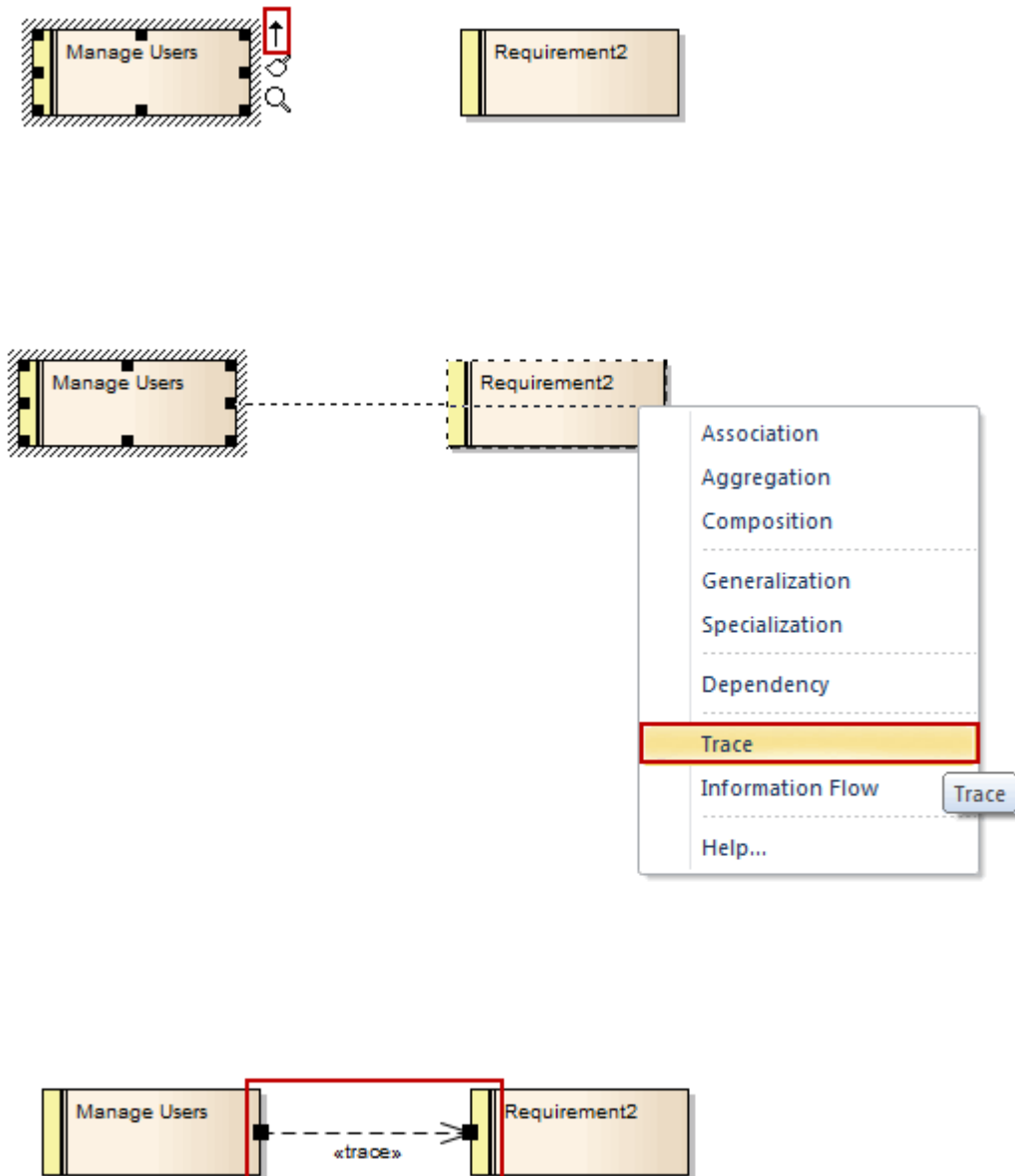


To access the Properties dialog, press **(Alt + Enter)**.

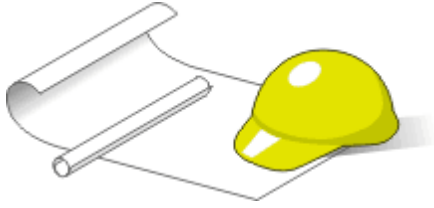


2.3.13 Requirement Trace Connector

Use the Quick Linker to create an appropriate connector between two Requirement elements



2.4 Typical Project Roles



Enterprise Architect performs a number of tasks that are suited to a variety of professions. This topic describes some common working practices with Enterprise Architect for a range of project roles. There are tools for the roles of:

Topic	Link
Business Analyst, to create high-level models of business processes	Business Analyst ^[74]
Software Architect, to map functional requirements, perform real time modeling of objects, design the Deployment model and detail the deliverable components	Software Architect ^[75]
Software Engineer, to map Use Cases onto Class diagrams, detail the interactions between Classes, define the system deployment and define software packages	Software Engineer ^[77]
Developer, to perform round trip code engineering, including reverse engineering of existing code and generation of code from Class elements	Developer ^[78]
Project Manager, to assign resources to elements, measure risk and effort, estimate project sizes, and manage element status, change control and maintenance	Project Manager ^[80]
Tester, to create test scripts against elements in the modeling environment	Tester ^[80]
Implementation Manager, to: <ul style="list-style-type: none"> Track and assign maintenance-related items to elements within Enterprise Architect Rapidly capture and keep records of maintenance tasks such as issues, changes, defects and tasks Trace the maintenance of the items and processes involved in system deployment 	Implementation Manager ^[81]
Technology Developer, to create customized additions to the functionality	Technology Developer ^[82]

Topic	Link
already present within Enterprise Architect	
Database Developer, to develop databases, including modeling database structures, importing database structures from an existing database and generating DDL for rapidly creating databases from a model	Database Developer ^[84]

You can review a summary of the **typical tasks** supported for each role, or click on the appropriate role title to explore how Enterprise Architect can assist you in carrying out that role within a model driven project.

Learn more

- [Summary of Typical Tasks](#)^[72]

2.4.1 Summary of Typical Tasks

Throughout a design and development project there are many different tasks to be performed, which could be carried out either by one person or - more probably - by members of a team with different responsibilities. In either case, Enterprise Architect supports most - if not all - of the responsibilities you might have on your project. The descriptions below cover a number of job roles that Enterprise Architect supports. For those that most resemble your role on a project, follow the job title hyperlink to display a description of how that role might use Enterprise Architect, then follow links within those topics to explore some of the Enterprise Architect features of importance to the role.

Summary of Typical Job Roles

Role	Responsibilities	See also
Business Analyst	For modeling: <ul style="list-style-type: none"> • Requirements • High-level business processes • Business activities • Work flows • System behavior 	Business Analyst ^[74]
Database Developer	<ul style="list-style-type: none"> • Developing databases • Modeling database structures • Creating logical data models • Generating schema • Reverse engineering databases 	Database Developer ^[84]
Software Architect	<ul style="list-style-type: none"> • Mapping functional requirements of the system 	Software Architect ^[75]

Role	Responsibilities	See also
	<ul style="list-style-type: none"> • Mapping objects in real time • Mapping the deployment of objects • Defining deliverable components 	
Tester	<ul style="list-style-type: none"> • Developing test cases • Importing requirements, constraints and scenarios • Creating Quality Test documentation • Tracking element defects and changes 	Tester [80]
Software Engineer	<ul style="list-style-type: none"> • Mapping Use Cases into detailed Classes • Defining the interaction between Classes • Defining system deployment • Defining software packages and the software architecture 	Software Engineer [77]
Project Manager	<ul style="list-style-type: none"> • Providing project estimates • Resource Management • Risk Management • Maintenance Management 	Project Manager [80]
Developer	<ul style="list-style-type: none"> • Forward, reverse and round-trip engineering • Visualizing the system states • Visualizing package arrangements • Mapping the flow of code 	Developer [78]
Implementation Manager	<ul style="list-style-type: none"> • Modeling the tasks in rolling-out a project, including network and hardware deployment • Assigning and tracking maintenance items on elements (issues, changes, defects and tasks) 	Implementation Manager [81]
Technology Developer	<p>For creating or customizing:</p> <ul style="list-style-type: none"> • UML Profiles • UML Patterns • Code Templates 	Technology Developer [82]

Role	Responsibilities	See also
	<ul style="list-style-type: none"> • Tagged Value Types • MDG Technologies • Add-Ins 	

Most of these roles work with specific types of diagram, so you might want to learn more about diagram types in general and specific types of diagram in particular.

Several types of project team member might want to generate documentation on their work and reports on how the project is developing and changing. Enterprise Architect enables you to generate project documentation in either RTF or HTML format.

Notes

- The Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect have a user security feature that can be applied or turned off. If security is turned on, you require the appropriate access permissions to use many of the Enterprise Architect facilities listed above

Learn more

- [List of Available Permissions](#) ^[264]
- [UML Diagrams](#) ^[1010]
- [Report Generation](#) ^[2249]

2.4.2 Business Analysts

A Business Analyst can use Enterprise Architect to create high-level models of business processes. These include business requirements, activities, work flow, and the display of system behavior.

Using Enterprise Architect, a Business Analyst can describe the procedures that govern what a particular business does. Such a model is intended to deliver a high-level overview of a proposed system.

Business Analyst Tasks

Topic	Detail	See also
Model High Level Business Processes	<p>With Enterprise Architect the Business Analyst can model high level processes of the business with Analysis diagrams.</p> <p>Analysis diagrams are a subset of UML 2.3 Activity diagrams and are less formal than other diagram types, but they provide a useful means for expressing essential business characteristics and requirements.</p>	Analysis Diagram ^[1509] Example Analysis Diagram ^[1511]
Model Requirements	Gathering requirements is typically the first step in developing a solution, be it for developing a software	Requirements Management ^[1468]

Topic	Detail	See also
	<p>application or for detailing a business process; it is an important step in the implementation of a project.</p> <p>Enterprise Architect enables you to define the Requirement elements, connect Requirements to the model elements for implementation, connect Requirements together into a hierarchy, report on Requirements, and move Requirements out of model element responsibilities.</p>	
Model Business Activities	<p>The Business Analyst can use Activity diagrams to model the behavior of a system and the way in which these behaviors are related to the overall flow of the system.</p> <p>Activity diagrams do not model the exact internal behavior of the system but show instead the general processes and pathways at a high level.</p>	Activity Diagram ^[1028]
Model Work Flow	<p>To visualize the cooperation between elements involved in the work flow, the Business Analyst can use an Interaction Overview diagram, which provides an overview of sub activities that are involved in a system.</p>	Interaction Overview Diagram ^[1087]
Display System Behavior	<p>In displaying the behavior of a system as a Use Case diagram, Enterprise Architect gives the Business Analyst an easily understood tool for mapping the functional requirements and behavior of a system.</p>	Use Case Diagram ^[1030]

Learn more

- [Business Modeling](#) ^[1519]

2.4.3 Software Architects

Software Architects can use Enterprise Architect to map functional requirements with Use Cases, perform real time modeling of objects using *Interaction diagrams* (Sequence, Timing, Communication or Interaction Overview), design the Deployment model and detail the deliverable components using Component diagrams.

Software Architect Roles

Topic	Detail	See also
Map Functional Requirements of the System	<p>With Enterprise Architect the Software Architect can take the high level business processes that have been modeled by the Business Analyst and create detailed Use Cases.</p>	Use Cases ^[1174]

Topic	Detail	See also
	Use Cases describe the proposed functionality of a system and are only used to detail a single unit of discrete work.	
Map Objects in Real Time	<p>The Software Architect can use Interaction diagrams (Sequence and Communication diagrams) to model the dynamic design of the system.</p> <p>Sequence diagrams are used to detail the messages that are passed between objects and the lifetimes of the objects.</p> <p>Communication diagrams are similar to Sequence diagrams, but are used instead to display the way in which the object interacts with other objects.</p>	<p>Sequence Diagram ^[1074]</p> <p>Example Sequence Diagram ^[1076]</p> <p>Timing Diagram ^[1051]</p> <p>Communication Diagram ^[1084]</p> <p>Interaction Overview Diagram ^[1087]</p>
Map Deployment of Objects	<p>The Software Architect can use Deployment diagrams to provide a static view of the run-time configuration of processing nodes and the components that run on the nodes.</p> <p>Deployment diagrams can be used to show the connections between hardware, software and any middleware that is used on a system.</p>	<p>Deployment Diagram ^[1020]</p>
Detail Deliverable Components	<p>Component diagrams enable the Software Architect to model the physical aspects of a system.</p> <p>Components can be executables, libraries, data files or another physical resource that is part of a system.</p> <p>The component model can be developed from scratch from the Class model or can be brought in from existing projects and from third-party vendors.</p>	<p>Component Diagram ^[1023]</p>

Learn more

- [Analysis Diagrams](#) ^[1509]
- [Modeling Fundamentals](#) ^[657]
- [XML Import and Export](#) ^[417]
- [XML Technologies](#) ^[2027]

2.4.4 Software Engineers

Software Engineers using Enterprise Architect can map Use Cases onto Class diagrams, detail the interactions between Classes, define the system deployment with Deployment diagrams and define software packages with Package diagrams.

Software Engineering Tasks

Topic	Detail	See also
Map Use Cases into Detailed Classes	<p>In Enterprise Architect the Software Engineer can study the Use Cases developed by the Software Architect, and with that information create Classes that fulfill the objectives defined in the Use Cases.</p> <p>A Class is one of the standard UML constructs that is used to detail the pattern from which objects are produced at run time; to record the relationships between Use Cases and Classes, the Software Engineer can create diagrams linking the elements with Realize connectors, and/or map the Realize connectors in the Relationship Matrix.</p>	Use Cases ^[1030] Example Use Case Diagram ^[1032] Class Diagram ^[1014] Realization ^[1254] Relationship Matrix ^[639]
Detail Interaction Between Classes	<p>Interaction diagrams (Sequence and Communication diagrams) enable the Software Engineer to model the dynamic design of the system.</p> <p>Sequence diagrams are used to detail the messages passed between objects and the lifetimes of the objects.</p> <p>Communication diagrams are similar to Sequence diagrams, but are used instead to display the way in which objects interact with other objects.</p>	Sequence ^[1074] Communication ^[1084]
Define System Deployment	<p>Deployment diagrams can be used to provide a static view of the run-time configuration of processing nodes and the components that run on the nodes.</p> <p>Deployment diagrams can be used to show the connections between hardware, software and any middleware that is used on a system, to explain the connections and relationships of the components.</p>	Deployment ^[1020]
Define Software	<p>The Software Engineer can use Package diagrams to detail the software architecture.</p>	Package ^[1011]

Topic	Detail	See also
Packages	Package diagrams are used to organize diagrams and elements into manageable groups, declaring the dependencies.	

Learn more

- [Modeling Fundamentals](#) ^[65]

2.4.5 Developers

Developers can use Enterprise Architect to perform round trip code engineering, which includes reverse engineering of existing code and generation of code from Class elements.

Developer Tasks

Topic	Detail	See also
Round Trip Engineering	<p>Enterprise Architect gives the developer unparalleled flexibility, with the ability to round trip software from existing source code to UML 2.3 models and back again.</p> <p>Round trip engineering involves both forward and reverse engineering of code; keeping the model and code synchronized is an important aspect of this.</p>	<p>Code Engineering ^[1733]</p> <p>Update Package Contents ^[1774]</p> <p>Estimation ^[516]</p>
Reverse Engineering	<p>Enterprise Architect enables developers to reverse engineer code from a number of supported languages and view the existing code as Class diagrams, which illustrate the static design view of the system.</p> <p>Class diagrams consist of Classes and interfaces and the relationships between them; the Classes defined in UML Class diagrams can have direct counterparts in the implementation of a programming language.</p> <pre> classDiagram class Account class AddressBook { + name: String + getContacts(): Contact + getContacts(): Contact[] + getName(): String + insertContact(contact): void + setName(String): void } class Contact { # emailAddress: String # faxNumber: String # name: String # primaryContactMethod: String + getEmailAddress(): String + getFaxNumber(): String + getName(): String + getPrimaryContactMethod(): String + setEmailAddress(String): void + setFaxNumber(String): void + setName(String): void + setPrimaryContactMethod(String): void } class ContactGroup { # name: String + getName(): String + setName(String): void } Account "1" -- "0..*" AddressBook : +IsAccessedBy AddressBook "0..*" -- "0..*" Contact : #Uses AddressBook "0..*" -- "0..*" Contact : +IsContainedIn Contact "0..*" -- "0..*" AddressBook : -Contains ContactGroup "0..*" -- "0..*" Contact : +Contains ContactGroup "0..*" -- "0..*" ContactGroup : +GroupedBy ContactGroup --> ContactGroup : #Parent 1 ContactGroup --> ContactGroup : +Child 0..* </pre>	<p>Reverse Engineering ^[1790]</p> <p>Class Diagram ^[1014]</p> <p>Class Element ^[1181]</p> <p>Resources ^[449]</p>

Topic	Detail	See also
Forward Engineering	<p>As well as the ability to reverse engineer code, Enterprise Architect offers the developer the option of forward engineering code (code generation).</p> <p>This enables the developer to make changes to their model with Enterprise Architect and have these changes implemented in the source code.</p>	Code Engineering ^[1733] Risk ^[453] Risk Types ^[458]
Determine the System State	<p>To visualize the state of the system the developer can use State Machine diagrams to describe how elements move between states, classifying their behavior according to transition triggers and constraining guards.</p> <p>State Machine diagrams are used to capture system changes over time, typically being associated with particular Classes (often a Class can have one or more State Machine diagrams used to fully describe its potential states).</p>	State Machine ^[1032]
Visualize Package Arrangement	<p>Package diagrams are used to help design the architecture of the system; they are used to organize diagrams and elements into manageable groups, and to declare their dependencies.</p>	Package ^[1011]
Follow the Flow of Code	<p>Activity diagrams are used to enable a better understanding of the flow of code.</p> <p>Activity diagrams illustrate the dynamic nature of the system, which enables modeling of the flow of control between Activities and represents the changes in state of the system.</p>	Activity ^[1028]

Notes

- State Machine, Package and Activity diagrams can be used by the developer to better understand the interaction between code elements and the arrangement of the code

Learn more

- [Model Transformations - MDA](#)^[1679]
- [Visual Execution Analysis](#)^[2143]

2.4.6 Project Managers

Enterprise Architect provides support for the management of projects. Project Managers can use Enterprise Architect to assign resources to elements, measure risk and effort, and estimate project sizes. Enterprise Architect also helps them manage element status, change control and maintenance.

Project Manager Tasks

Topic	Detail	See also
Provide Project Estimates	With Enterprise Architect the Project Manager has access to a comprehensive project estimation tool that calculates effort from Use Case and Actor objects, coupled with project configurations defining the technical and environmental complexity of the work environment.	Estimation ^[516]
Resource Management	Managing the allocation of resources in the design and development of system components is an important and difficult task; Enterprise Architect enables the Project Manager or Development Manager to assign resources directly to model elements and track progress over time.	Resources ^[449]
Risk Management	The Project Management window can be used to assign Risk to an element within a project; the risk types enable the Project Manager to name the risk, define the type of risk, and give it a weighting.	Risk ^[453] Risk Types ^[458]
Maintenance	Enterprise Architect enables the Project Manager to track and assign maintenance-related items to elements within Enterprise Architect; this enables rapid capture and record keeping for items such as issues, changes, defects, and tasks. A Project Manager can also create and maintain a project glossary of processes, procedures, terms and descriptions.	Issues ^[462] Changes ^[2244] Defects ^[2244] Tasks ^[460] Project Glossary ^[467]

Learn more

- [Projects Management](#) ^[447]
- [Element Status](#) ^[529]
- [Maintenance](#) ^[2238]

2.4.7 Testers

Enterprise Architect provides support for design testing by enabling you to create test scripts against elements in the modeling environment.

Testing Tasks

Topic	Detail	See also
Test Cases	<p>With Enterprise Architect, Quality Assurance personnel can set a series of tests for each model element.</p> <p>The test types include Unit testing, Acceptance testing, System testing and Scenario testing.</p>	
Import requirements, constraints and scenarios	<p>To help ensure that testing maintains integrity with the entire business process, Enterprise Architect enables the tester to import requirements, constraints and scenarios defined in earlier iterations of the development life cycle.</p> <p>Requirements indicate contractual obligations that elements must perform within the model.</p> <p>Constraints are conditions that must be met in order to pass the testing process; constraints can be:</p> <ul style="list-style-type: none"> • Pre-conditions (states which must be true before an event is processed) • Post Conditions (events that must occur after the event is processed) or • Invariant constraints (which must remain true through the duration of the event) <p>Scenarios are textual descriptions of an object's action over time and can be used to describe the way a test works.</p>	<p>Requirements^[1468]</p> <p>Constraints^[850]</p>
Create quality test documentation	Enterprise Architect provides the facility to generate high quality test documentation in the industry-standard .RTF file format.	
Element defect changes	Defect tracking enables you to allocate defect reports to any element within the Enterprise Architect model; this enables all who are involved in the project to quickly view the status of defects, to see which defects have to be addressed and which have been dealt with.	Element Defects ^[2238]

Learn more

- [Introduction to Testing](#)^[2217]

2.4.8 Implementation Managers

Enterprise Architect provides support for the management of project implementation. Enterprise Architect enables you to track and assign maintenance-related items to elements within Enterprise Architect. This enables you to rapidly capture and keep records of maintenance tasks such as issues, changes, defects and tasks. By providing a centralized facility for each element involved in the deployment process Enterprise Architect offers a powerful solution for tracing the maintenance of the items and processes involved in system deployment.

Implementation Tasks and Tools

Topic	Detail	See also
Deployment Diagram	<p>Using Deployment diagrams in Enterprise Architect, you can model the roll out of a project, including network deployment and workstation deployment.</p> <p>Users involved in project deployment can add maintenance tasks to the diagram elements.</p> <p>Deployment diagrams provide a static view of the run-time configuration of nodes on the network or of workstations, and the components that run on the nodes or are used in the workstations.</p>	<p>Deployment Diagram ^[1020]</p> <p>Example Deployment Diagram ^[1021]</p>

Learn more

- [Project Maintenance](#) ^[2237]

2.4.9 Technology Developers

Technology Developers are Enterprise Architect users who create customized additions to the functionality already present within Enterprise Architect.

Additions include UML Profiles, UML Patterns, Code Templates, Tagged Value Types, Scripts, Custom Queries, Transformations, MDG Technologies and Enterprise Architect Add-Ins. By creating these extensions the Technology Developer can customize the Enterprise Architect modeling process to specific tasks and speed up development.

Developing Technologies

Topic	Detail	See also
UML Profiles	<p>By creating UML Profiles the technology developer can create a customized extension for building UML models that are specific to a particular domain.</p> <p>Profiles are stored as XML files and can be imported into any model as required.</p>	UML Profiles ^[1277]
UML Patterns	<p>Patterns are sets of collaborating Objects and Classes that provide a generic template for repeatable solutions to modeling problems.</p> <p>As patterns are discovered in any new project, the basic pattern template can be created.</p> <p>Patterns can be re-used with the appropriate variable names modified for any future project.</p>	Design Patterns ^[1277]
Code Templates	Code templates are used to customize the output of source code generated by Enterprise Architect; this enables the	Code Templates ^[1924]

	generation of code languages not specifically supported by Enterprise Architect and enables you to define the way Enterprise Architect generates source code to comply with your own company style guidelines.	
Tagged Value Types	<p><i>Tagged Values</i> are used in Enterprise Architect to extend the information relating to an element in addition to the information directly supported by the UML language.</p> <p>A Tagged Value, strictly, is the value of a property of a modeling item, the property being called a tag; for example: a Class element called Person might have a tag called Age with the Tagged Value of 42.</p> <p>More loosely, the combination of tag and value can be referred to as a Tagged Value.</p> <p>A Tagged Value type is a group of parameters that define and/or limit the possible values of a tag and, in many instances, how a specific value is assigned to the tag; for example, the tag Age might have a Tagged Value Type of Integer, so the user simply types in a numeric value.</p> <p>Alternatively, the type could be Spin, with lower and upper limits of, say, 20 and 120, so the user sets a value by clicking on arrows in the field to increment or decrement the value within the limits of 20 and 120.</p> <p>Typically, Tagged Values are used during the code generation process, or by other tools to pass on information that is used to operate on elements in particular ways.</p>	<p>Tagged Values ^[970]</p> <p>Tagged Value Type ^[985]</p>
MDG Technologies	MDG Technologies can be used to create a logical collection of resources that can contain UML Profiles, Patterns, Code Templates, Image files and Tagged Value types that are accessed through a technology file.	MDG Technologies ^[1283]
Enterprise Architect Add-Ins	Add-Ins enable you to build your own functionality into Enterprise Architect, creating your own mini programs that can extend the capabilities of Enterprise Architect, defining your own menus, and creating your own Custom Views.	Enterprise Architect Add-Ins ^[2585]

Learn more

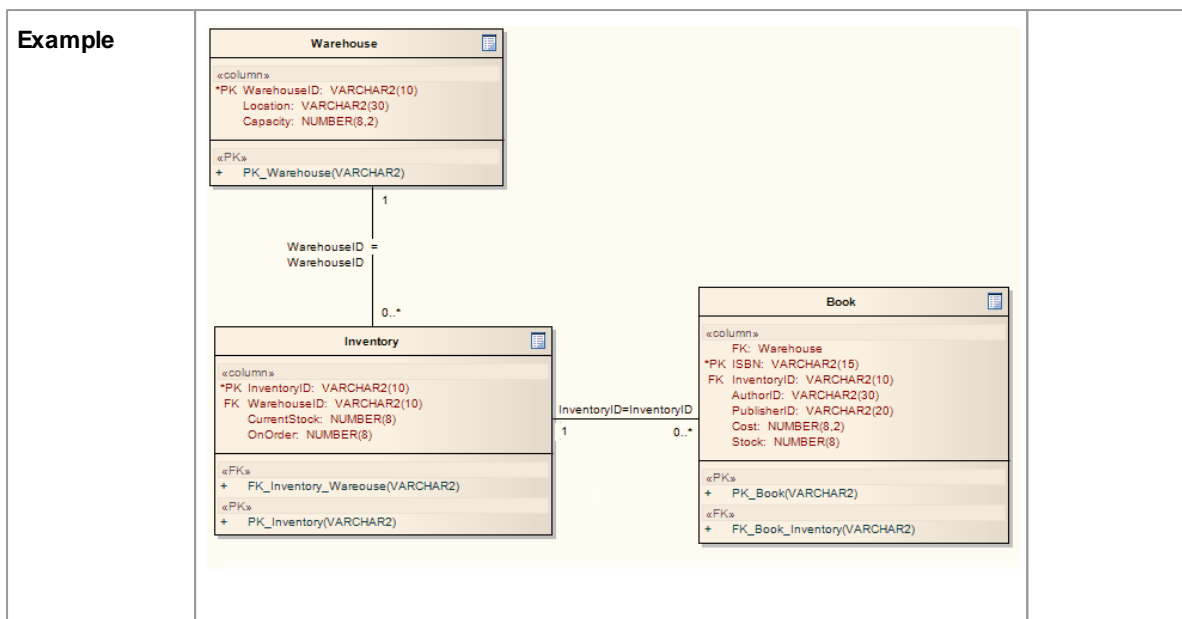
- [Extending UML](#) ^[1292]
- [MDG Technology SDK](#) ^[1292]

2.4.10 Database Developers

Enterprise Architect supports a range of features for the development of databases, including modeling database structures, importing database structures from an existing database and generating DDL for rapidly creating databases from a model.

Database Development Tasks

Topic	Detail	See also
Create Logical Data Models	<p>With Enterprise Architect the Database developer can build database diagrams using the built-in UML Data Modeling Profile.</p> <p>This supports the definition of Primary and Foreign keys, cardinality, validation, triggers, constraints and indexes.</p>	Logical Data Model ^[1627]
Generate Schema	<p>By using Enterprise Architect's DDL generation function the Database Administrator can create a DDL script for creation of the database table structure from the model.</p> <p>Enterprise Architect currently supports:</p> <ul style="list-style-type: none"> • JET-based databases • DB2 • InterBase • Informix • Ingres • MySQL • SQL Server • PostgreSQL • Sybase Adaptive Server Anywhere and Adaptive Server Enterprise, and • Oracle 9i, 10g and 11g 	Database Schema ^[1637]
Reverse Engineer Database	<p>Using an ODBC data connection the Database Administrator can import a database structure from an existing database to create a model of the database.</p> <p>Generating the model directly from the database enables the DBA to quickly document their work and create a diagrammatic account of a complex database through the graphical benefits of UML.</p>	Reverse Engineering ^[1796]



Learn more

- [Data Models](#) ^[1625]
- [Physical Data Model](#) ^[1932]

2.5 User Interface Guide



This section describes the *Enterprise Architect Application Workspace*, which provides a powerful and flexible analysis, modeling, design and construction environment. The workspace is built from familiar Windows components, such as toolbars, docking windows, dialogs and menus. Together these elements provide a **rich set of tools** for working with diagrams, models, **UML**, source code, executing applications (debugging and visualization), rich text documents, collaborative tools, extension technologies and much more.

User Interface Guide

Topic	Link
The Enterprise Architect workspace, the interface through which you create and display your models; the main workspace component is the central Diagram View , which is where you create model diagrams, view reports, scroll through lists of model elements, access the internet and even edit and debug source code	The Start Page ^[87] Diagram View ^[69] The Web Browser ^[165]
The Main Menu , which provides access both to high-level functions related to the project life cycle, and to project and system administration functions; throughout Enterprise Architect you can also access functions and operations using context menus	The Main Menu ^[90] Project Browser Context Menus ^[569] Diagram Context Menu ^[687]
The windows that are used to enter and display information on your project and models; you can rearrange windows to adapt the screen space to your work habits	Standard Windows ^[132]
Toolbars are small sets of buttons that provide quick access to perform common operations both on your project as a whole and on the individual modeling units	Standard Toolbars ^[138]
The Enterprise Architect Diagram Toolbox provides all the components and connectors that you use to create models using whatever diagrams are appropriate; the Toolbox automatically matches the kind of diagram you have open, and whichever technology is currently active in your model	The Diagram Toolbox ^[698]
The key combinations that you can use throughout Enterprise Architect to quickly initiate a wide range of actions	Keyboard Shortcuts ^[166]

Topic	Link
The customization tools that enable you to tailor the menus, toolbars and function keys that you use in Enterprise Architect, to set the visual style of the application, and to customize the workspace layout	Customization ^[153] Visual Styles ^[162] Manage Workspace Layout ^[163]
The navigation and search tools , which enable you to search for, track and locate information in your project, and filter the information so that you can explore particular views of your project rather than the entire contents	Navigate, Search and Trace ^[566] The Project Browser ^[567]

Learn more

- [A Demonstration of Enterprise Architect in use](#)

2.5.1 Start Page

The Start Page provides some useful navigation tools and jumping off points when you first open Enterprise Architect - including the **recent files list**, commands to **create or copy projects**, connect to **repository servers** and useful **Help topics** to get you started. Links to online resources and to the Sparx Systems Enterprise Architect Community website can also help you get up to speed and explore the possibilities of Enterprise Architect.

If you have no projects of your own to open yet, open the *EAExample* project or use the *QuickStart Tutorial* to create a project to explore.

Using the Start Page

Action	Usage	See also
Workspace Layout	Apply a workspace layout. By selecting a named workspace layout from the drop list provided, you can quickly re-arrange the windows in Enterprise Architect into a specific pre-determined form. Each layout is generally named according to the purpose it is intended for.	Manage Workspace Layout ^[163]
Open a Project File	Display the Select Enterprise Architect Project to Open dialog, which you use to open an existing project (where you have more project files than can be listed in the Recent panel).	Open a Project ^[187]
Create a New Project	Save a new project and open the Model Wizard dialog.	Create a New Project ^[189] Model Wizard ^[660]
Copy a Base Project	Select a different Base Project to generate a new project from.	Copy a Base Project ^[190]

Action	Usage	See also
Connect to Server	<p>Specify a Data Source name to connect to; the supported repositories are listed on the right.</p> <p>This feature is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions.</p>	<p>Connect to Server^[209]</p> <p>MySQL^[195]</p> <p>SQL Server^[195]</p> <p>Oracle 9i, 10g and 11g^[197]</p> <p>PostgreSQL^[197]</p> <p>MSDE^[198]</p> <p>Adaptive Server Anywhere^[198]</p> <p>Access 2007^[194]</p> <p>Progress OpenEdge^[199]</p>
Getting Started	Open the Learning Center, to display useful topics and guides for various areas of work in Enterprise Architect.	Learning Center ^[89]
Configure Options	Display the Help on the Options dialog, which enables you to define how Enterprise Architect displays and processes information.	Options ^[542]
Online Resources & Tutorials	Open the Resources page of the Sparx Systems website, which provides access to a wide range of Enterprise Architect and UML tutorials, demonstrations, examples, Add-Ins and advice.	The Web Browser ^[165]
Enterprise Architect Community	<p>Access the Sparx Systems Enterprise Architect Community Site, which contains a range of articles, discussions, tools and resources provided by both Sparx Systems and the Enterprise Architect user community.</p> <p>You must register to use the facilities of the site; you can also register as an author and submit material yourself, for others to read and use.</p>	Sparx Systems Enterprise Architect Community Site
Recent	<p>Select from a list of hyperlinks to the most recently used Enterprise Architect projects (both .EAP files and DBMS connections).</p> <p>If necessary, you can remove the hyperlink to a project from this list.</p>	Remove Recent Projects ^[89]

Notes

- If your model has a default diagram set, that diagram opens immediately, hiding the Start Page; you can still access the Start Page from the diagram tabs below the diagram

- If you have set a shortcut view profile, that overrides the default diagram setting
- You can also hide the Start Page, and show it again, by de-selecting and selecting the **Window | Show Start Page** option; the setting persists though shut down and restart, until you specifically change the option again

Learn more

- [Quickstart Tutorial](#) ^[49]
- [Set the Default Diagram](#) ^[74]
- [Diagram Tabs](#) ^[69]

2.5.1.1 Remove Recent Projects

Removing the hyperlink to a project from the Start Page only removes the link to the project and does not remove the project from the file system or repository.

Access **File | Open Project > Open Enterprise Architect Project > Recent Projects: Remove Selection from List** (Ctrl+O)

How to remove a project hyperlink from the Recent list on the Start Page

Step	Action	See also
1	On the Open Enterprise Architect Project dialog, in the Recent Projects panel, click on the project to remove.	
2	Click on the Remove Selection from List button.	

2.5.2 Learning Center

The Learning Center window provides quick access to a range of context-specific Enterprise Architect help topics, online resources and facilities in a number of specific work areas in Enterprise Architect, such as Getting Started, Modeling or Testing.

These topic areas can include topics specific to any MDG Technologies being used with Enterprise Architect.

Access **View | Learning Center** (Ctrl + Shift + 9)

(Also, when you first open Enterprise Architect, the Learning Center automatically displays on the right of the screen)







Uses

- Initiate a task or process within a specific work area
- Display topics from Enterprise Architect Help that describe either procedures or facilities to achieve objectives in a work area

- Display web pages or documents that provide conceptual information or examples within a work area
- Run selected reports or data collation tools
- Run demonstrations of functions within a work area

To display information listed in the Learning Center, click on the required item.

The types of item are indicated by the list titles and by the following icons

Icon	Action
	Open appropriate topics from the Enterprise Architect Help file.
	Open web pages or documents on the Sparx Systems web site.
	Begin Enterprise Architect tasks appropriate to the Learning Center topic area; you must be in an appropriate functional area of Enterprise Architect in order for these tasks to function, such as in an open diagram.
	Begin Add-In tasks appropriate to the Learning Center topic area; you must be in an appropriate functional area in order for these tasks to function.
	Open report facilities to provide information or data collation tools.
	Start demonstrations of Enterprise Architect functions in action.

The selected information, web page or demonstration displays on a Browser tab in the main view, or the appropriate task or report window opens.

To switch between the topic areas, either:

- Click on the drop-down arrow at the right of the Learning Center toolbar and select the required area from the list, or
- Click on the left or right arrow buttons in the toolbar

The 'Home' icon returns you to the Getting Started topic area.

2.5.3 Main Menu

The Enterprise Architect **Main Menu**, at the top of the Enterprise Architect workspace, opens a range of subordinate menus that provide mouse-driven access to many high-level functions related to the project life cycle, as well as administration functions.

Each subordinate menu is described in the following table

Usage	See also
File - provides options to create, open, close and save projects, and also to perform print tasks.	File ^[92]
Edit - provides a range of functions to apply to diagram elements for the currently open diagram or view.	Edit ^[93]
View - enables you to display each of the windows or toolbars, and to set the visual style of your Enterprise Architect workspace.	View ^[95]
Project - provides access to tasks related to the management of your project, such as adding to the project structure, importing and exporting data, generating documentation, version control and security.	Project ^[100]
Diagram - enables you to configure diagram properties and options and to save diagram images to file.	Diagram ^[105]
Element - provides options to configure and access element features and details and control element layout, and to generate documentation and work on source code for the element.	Element ^[108]
Tools - provides access to tools related to code engineering, model management, spell checking, customization of features and setting operational options.	Tools ^[115]
Analyzer - provides access to a range of functions to perform build, debug, simulation and profiling operations on your model code.	Analyzer ^[120]
Extensions - enables you to connect to, display information on, work with and manage your Add-Ins.	Extensions ^[125]
Settings - enables you to configure various settings for your overall project, such as stereotypes, Tagged Values, cardinality values, datatypes, language macros, local directories and image management.	Settings ^[127]
Window - provides various functions for managing open system windows.	Window ^[129]
Help - provides access to the Enterprise Architect help files, the Read Me file, the Enterprise Architect End User License Agreement, the Enterprise Architect Example model and various features on the Sparx Systems website; it also enables you to manage your license keys.	Help ^[131]

Additionally, if you right-click on the **Toolbar** area just under the menu bar, a composite context menu

displays providing options to display the toolbars and all significant windows and views.

2.5.3.1 File

The **File** menu provides options to create, open, close and save projects, and also to perform print tasks.

Access **File**

The File Menu

Option	Usage	Shortcut	See also
New Project	Create a new Enterprise Architect project.	Ctrl + N	File Based Repositories ^[189]
Open Project	Open a project.	Ctrl + O	Open a project ^[181]
Reload Current	Reload the current project (use in a multi-user environment to refresh the Project Browser).	Ctrl + Shift + F11	Refresh View of Shared Project ^[242]
Close Project	Close the current project.		
Save Project Copy	Save the current model with a new name.		Copy Existing Project ^[191]
Save Project Shortcut	Create a desktop shortcut to the current model. (This option is also active in the 'Lite', read-only edition of Enterprise Architect.)		Project Shortcuts ^[183] The Read-Only 'Lite' Edition ^[261]
Print Setup	Configure your printer's settings.		
Page Setup	Configure the page settings to print the current diagram.		Set Diagram Page Size ^[764]
Print Preview	Preview ^[690] how the currently displayed diagram prints.		Print Preview ^[690]
Print	Print the currently displayed diagram. Enterprise Architect provides facilities to change the scale of the printed diagram (the number of pages it takes up) and to print or omit page headers and footers on the diagram.	Ctrl + P	Scale Image To Page Size ^[766] Diagram Tab ^[729]
Recent Files List	Select from a list of the most recently opened projects.		

Option	Usage	Shortcut	See also
Exit	Exit Enterprise Architect.		

2.5.3.2 Edit

The **Edit** menu provides a range of functions to apply to diagram elements for the currently open diagram or view.

Access [Edit](#)

The Edit Menu

Option	Usage	Shortcut	See also
Model Search	Display the Model Search window, to search for particular phrases or words.	Ctrl + Alt + A	Model Search ^[61]
File Search	Display the File Search window, to search for text in code files and scripts.	Ctrl + Shift + Alt + F	Find In Files ^[181]
Copy	<ol style="list-style-type: none"> Copy the selected elements to the MS Windows clipboard. (To paste the selected elements use the Paste Element(s) option, below.) Copy an image of the selected elements to the clipboard. If no elements are selected, the entire diagram is copied. <p>The image can be saved as a bitmap or a metafile; you set the format on the Options dialog.</p>	Ctrl + C	General ^[54] (page of the Options dialog)
Project Clipboard	<p>Offers two sub-options:</p> <ul style="list-style-type: none"> Add to Project Clipboard - Add the current element to the Enterprise Architect clipboard Clear Project Clipboard - Clear any elements from the Enterprise Architect clipboard 	Ctrl + Space ---	
Paste Element (s)	Display the Paste Elements submenu, which enables you to paste elements on the clipboard		Paste Elements Submenu ^[94]

Option	Usage	Shortcut	See also
	into the current diagram.		
Paste Image From Clipboard	Paste the element in the Enterprise Architect Clipboard into the same diagram or a different diagram, as a metafile (that is, a definition of the element) as many times as is necessary. If you paste the element into a different diagram, its classifier identifies the source diagram for the element.	Ctrl + Shift + Insert	
Redo	Re-apply the last undone action.	Ctrl + Y	Redo Last Action ^[768]
Undo	Undo the last action performed; note that some actions cannot be undone.	Ctrl + Z	Undo Last Action ^[768]
Select All	Select all elements concurrently on the current diagram.		
Select By Type	Specify a particular type of element to select on the diagram.		
Clear Selection	Deselect all elements.		
Bookmark Selected	Bookmark the selected element(s). If the selected element is already bookmarked, this option removes the bookmark.	Shift + Space	Manage Bookmarks ^[530]
Clear All Bookmarks	Clear bookmarks from any bookmarked elements in the current diagram.		Manage Bookmarks ^[530]
Delete Selected Element(s)	Delete the selected elements from the diagram.	Ctrl + D	

2.5.3.2.1 Paste Elements Submenu

You can paste an element from the Enterprise Architect Clipboard or the MS Windows clipboard into a diagram, either as a hyperlink to the original element or as a new element. The Enterprise Architect clipboard takes precedence, so you must clear that clipboard before you can paste from the MS Windows clipboard.

Access [Edit | Paste Element\(s\)](#)

Paste Elements Menu

Option	Usage	Shortcut	See also
as Link	<p>Paste the element in the buffer as a link (that is, a reference) to the element.</p> <p>If there are images in the MS Windows clipboard and none in the Enterprise Architect clipboard, you can:</p> <ul style="list-style-type: none"> • Paste an image from the MS Windows clipboard into a new element as the appearance of the new element, or • Paste an image from the MS Windows clipboard into the diagram as a new boundary's appearance 	Shift + Insert	Paste from Project Browser ^[733]
as New	Paste the element in the buffer as a completely new ^[724] element.	Ctrl + Shift + V	

2.5.3.3 View

The **View** menu enables you to display and/or focus on each of the windows or toolbars, and to set the visual style of your Enterprise Architect workspace.

Access View**The View Menu**

Option	Usage	Shortcut	See also
Workspace Layouts	Show the Workspace Layout dialog for changing the content and layout of the Enterprise Architect workspace.		Manage Workspace Layout ^[163]
Learning Center	Show the Learning Center, providing context-specific help topics, online resources and facilities in a number of specific work areas in Enterprise Architect.	Ctrl + Shift + 9	Learning Center ^[89]
Project Browser	Display the Project Browser, for browsing and exploring your model; this displays the hierarchical arrangement of packages, elements and features within the model.	Alt + 0	The Project Browser ^[567]

Option	Usage	Shortcut	See also
Package Browser	Display the elements of the current package in the Package Browser, as a context-sensitive, editable table.	Ctrl + Alt + R	Package Browser ^[587]
Model Views	Show the Model Views window, through which you can create filtered views of selected areas or aspects of your model.	Ctrl + Shift + 5	Model Views ^[598]
More Project Tools	Display a submenu containing options for the Resources and Project Information windows, and the internal Web Browser.		View Submenus ^[98]
Personal Information	Display the Personal Information window, which enables you to email project team members and report on the work and tasks assigned to you.		Personal Information ^[486]
Team Review	Show the Team Review window, the portal for a discussion forum within your development team community.	Ctrl + Alt + U	Team Review Tools ^[279]
Task Allocations	Show the Project Task Allocation window, through which you manage work tasks allocated within the project and run reports on the components associated with the tasks.		Project Task Allocation ^[472]
Diagram Toolbox	Show the Diagram Toolbox, a panel of icons for creating elements and connectors on a diagram; there are pages of icons for each UML, Extended and MDG Technology diagram type.	Alt + 5	Diagram Toolbox ^[698]
Diagram Filters	Display the Diagram Filters window, for selecting the elements to show or hide on a diagram.		Diagram Filters ^[629]
Diagram Layout	Display the Layout Tools window, for reformatting your diagram in one of a range of layouts.		Layout Diagrams ^[769]
Pan and Zoom	Display the Pan & Zoom window for panning across a diagram to display sections at greater magnification.	Ctrl + Shift + N	The Pan & Zoom Window ^[610]
Element	Show the Properties window for the selected	Alt + 1	Properties Window

Option	Usage	Shortcut	See also
Properties	element, which provides an editable table of the common properties of the selected element.		[878]
Relationships	List a selected element's connectors on the Relationships window.	Ctrl + Shift + 2	The Relationships Window [648]
Traceability	Show the Traceability [637] window, for tracing the relationships of an element through the model.	Ctrl + Shift + 4	The Traceability Window [637]
More Element Tools	Display a submenu containing options for the Scenarios & Requirements, Element Browser, Project Management, Maintenance, Testing and Source Code windows.		View Submenus [98]
Relationship Matrix	Display the Relationship Matrix, to cross reference elements to each other by connector type.		Relationship Matrix [639]
Gap Analysis Matrix	Display the Gap Analysis Matrix to analyze model artifacts for potential gaps in the solution.		Gap Analysis Matrix [65]
Notes	Show the Notes window, to create and review the explanatory text on a diagram, element, feature or connector.	Ctrl + Shift + 1	Notes [97]
Tagged Values	Show the Tagged Values window, to add further properties to a modeling object beyond those supported by UML.	Ctrl + Shift + 6	Tagged Values [969]
System Output	Show the Output window	Ctrl + Shift + 8	The System Output Window [164]
Scripting	Show the Scripting window.		Scripting [2372]
Toolbars	Show or hide individual toolbars.		View Submenus [98]
Visual Style	Set the Visual Style and a number of other display options.		View Submenus [99]

2.5.3.3.1 View Submenus

The More Project Tools Sub-Menu

Select the windows to be visible and deselect those to be hidden. You can select from:

Option	Usage	Shortcut	See also
Project Resources	Show or hide the Resources window.	Alt+6	Resources [1003]
Project Information	Show or hide the Project Information window.	Alt+2	The Project Information Window [460]
Internal Web Browser	Open the web browser page at the site you have specified on the Options dialog, in the Web Home field.	Ctrl+Alt+W	Internal Web Browser [165] Options [544]

The More Element Tools Submenu

Select the windows to be visible and deselect those to be hidden. You can select from:

Option	Usage	Shortcut	See also
Scenarios & Requirements	Display the Scenarios, internal Constraints and Requirements for the selected element, as tabs of the main work space.	Ctrl+Shift+3	Scenarios, Internal Constraints and Requirements [879]
Element Browser	Explore the components of the selected element, in the Element Browser window.	Alt+9	Element Browser [876]
Project Management	Show or hide the Project Management window.	Ctrl+Shift+7	Project Management [448]
Maintenance	Show or hide the Maintenance window.	Alt+4	Maintenance [237]
Testing	Show or hide the Testing window.	Alt+3	Testing [2213]
Source Code	Show or hide the Source Code window.	Alt+7	Editing Source Code [1799]

The Toolbars Sub-Menu

Select the toolbars to be visible and deselect those to be hidden. You can select from:

Topic	Usage	Link
Default Tools	Show or hide the Default Tools toolbar.	Default Tools ^[139]
Project	Show or hide the Project toolbar.	Project ^[141]
Code Generation	Show or hide the Code Generation toolbar.	Code Generation ^[142]
UML Elements	Show or hide the UML Elements toolbar.	UML Elements ^[143]
Diagram	Show or hide the Diagram toolbar.	Diagram ^[145]
Current Element	Show or hide the Current Element toolbar.	Current Element ^[146]
Current Connector	Show or hide the Current Connector toolbar.	Current Connector ^[148]
Workspace Layouts	Show or hide the Workspace Layouts toolbar.	Workspace Layouts ^[149]
Debug & Record	Show or hide the Debug & Record toolbar.	Debug & Record ^[150]
Status Bar	Show or hide the Status Bar.	Status Bar ^[152]

The Visual Style Sub-Menu

Option	Usage	Shortcut	See also
Select Visual Style	Select different visual styles or <i>themes</i> for the Enterprise Architect user interface.		Visual Styles ^[162]
Animate Autohide Windows	Animate windows that have been automatically hidden.		Automatically Hidden ^[137]
Show Toolbar Customize Button	Toggle the down-arrow on the end of each toolbar that enables you to customize the toolbar buttons, as shown below:		

Option	Usage	Shortcut	See also
Hide Diagram Caption Bar	Hide or redisplay the diagram caption bar at the top or bottom of a diagram.		Caption Bar ^[69]

2.5.3.4 Project

The **Project** menu gives access to tasks related to the management of your project, such as adding to the project structure, importing and exporting data, generating documentation, version control and security.

Access **Project**

The Project Menu

Action	Usage	Shortcut	See also
New Model	Display the Model Wizard, to add models to your project.		Model Wizard ^[66]
New Package	Create a new package.	Ctrl + W	Create a new package ^[68]
New Diagram	Create a new diagram in the current package.	Ctrl + Insert	Create a new diagram ^[72]
Project Calendar	Display the Calendar, to monitor work across the project in terms of the resources involved, the tasks being performed, the events occurring and the elements being worked on.		Project Calendar ^[50]
Documentation	Display the Documentation submenu.		Documentation Submenu ^[10]
Model Validation	Display the Model Validation submenu.		Model Validation Submenu ^[102]
Baselines	Store a model branch as a snapshot or baseline. Available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions.	Ctrl + Alt + B	Baseline ^[38]
Audit View	Display the Audit View, which shows the information that has been recorded by auditing.		Audit View ^[38]
Model Import/	Display the Import/Export submenu.		Import/Export

Action	Usage	Shortcut	See also
Export			Submenu ^[102]
Version Control	Display the Version Control submenu.		Version Control Submenu ^[103]
Security	Display the Security submenu.		Security Submenu ^[104]
Use Case Metrics	Set Use Case Metrics to assist in estimating project size.		Use Case Metrics ^[520]
Project Statistics	View some basic project statistics.		

2.5.3.4.1 Documentation Submenu

Access **Project | Documentation**

The Documentation Menu

Action	Usage	Shortcut	See also
Rich Text Format (RTF) Report	Generate a report for the currently selected package in Rich Text Format.	F8	RTF Documents ^[225]
HTML Report	Generate a report for the currently selected package in HTML format.	Shift + F8	HTML Reports ^[235]
Diagrams Only Report	Generate an RTF report containing only diagrams.	Ctrl + Shift + F8	Diagrams Only Report ^[234]
Testing Report	Generate an RTF report of the model's existing tests.		Test Documentation ^[223]
Issues	Generate an RTF report of the model's issues.		Project Issues ^[462]
Glossary	Generate an RTF report of the model's Glossary.		Generate a (Glossary) Report ^[47]
Implementation Details	Show a list of the elements in the package (selected in the Project Browser) that require implementers, together with any source elements in Realize (Implements) relationships with those elements.		Implementation Report ^[234]

Action	Usage	Shortcut	See also
Dependency Details	Show a list of any elements in the package (selected from the Project Browser) that are dependent on another element for their specification.		Dependency Report ^[233]
Testing Details	Show the testing details for the elements in a package, for tests that meet specific criteria.		Testing Details Report ^[234]

2.5.3.4.2 Model Validation Submenu

Access **Project | Model Validation**

The Model Validation Menu

Action	Usage	Shortcut	See also
Validate Selected	Validate a selected element, diagram or package from the Project Browser.	Ctrl + Alt + V	Model Validation ^[220]
Cancel Validation	Cancel the validation process.		
Configure	Configure the Validation rules from the list of available rules.		Configure Model Validation ^[220]

2.5.3.4.3 Import/Export Submenu

To perform import and export to XMI and CSV

Access **Project | Model Import/Export**

The Import/Export Menu

Action	Usage	Shortcut	See also
Import Package from XMI	Import a package from an XMI (XML based) file.	Ctrl + Alt + I	Import From XMI ^[41]
Export Package to XMI	Export the currently selected package to an XMI (XML based) file.	Ctrl + Alt + E	Export To XMI ^[41]

Action	Usage	Shortcut	See also
Batch XMI Export	Export a group of controlled packages in one action.		Batch XMI Export [430]
Batch XMI Import	Run a batch import of multiple packages.		Batch XMI Import [431]
CSV Import/Export	Import or export information on Enterprise Architect elements in CSV format.	Ctrl + Alt + C	CSV Import [440] CSV Export [439]
CSV Import/Export Specifications	Set up CSV import Export Specifications.		CSV Specifications [435]
Export Reference Data	Export reference data to XML files for convenient model updating.		Export Reference Data [306]
Import Reference Data	Import reference data from XML files for convenient model updating.		Import Reference Data [308]

2.5.3.4.4 Version Control Submenu

Access **Project | Version Control**

The Version Control Menu

Action	Usage	Shortcut	See also
Configure Current Package	Specify whether this package (and its children) is controlled and, if so, which file it is controlled through.	Ctrl + Alt + P	Configure Controlled Package [362]
Version Control Settings	Specify the options required to connect to a Source Code Control (SCC) provider.		Version Control Settings Dialog [346]
Validate Package Configurations	Validate the version control configuration of each package in the model.		Validate Package Configurations [379]
Resynch Statuses of All Packages	Resynchronize the status of all version controlled packages within your project with the status of the version control provider.		Resynchronize the Status of Version Controlled Packages [380]

Action	Usage	Shortcut	See also
Work Offline	Toggle between 'offline' version control and online version control.		Offline Version Control ^[322]

2.5.3.4.5 Security Submenu

To configure security settings for your project.

Access **Project | Security**

The Security Menu

Action	Usage	Shortcut	See also
Manage Users	Add, modify and remove users, including maintaining permissions.		Maintain Users ^[256]
Manage Groups	Add, modify and remove security groups, including maintaining permissions.		Maintain Groups ^[262]
Manage Locks	View and manage element locks.		View and Manage Locks ^[268]
Require User Lock to Edit	Toggle the security policy in force.		Security Policy ^[255]
Manage My Locks	View and delete your own locks.	Ctrl + Shift + L	Manage Your Own Locks ^[278]
Login as Another User	Switch the login to a different user ID.		
Change Password	Change current security password.		Change Password ^[270]
Encrypt Password	Add encryption to your password (prior to Enterprise Architect Release 7.1).		Password Encryption ^[269]
Enable Security	Enable or disable user security to limit access to update functions in the model.		Enable Security ^[253]

Notes

- This feature is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions

2.5.3.5 Diagram

The **Diagram** menu enables you to configure diagram properties and options and to save diagram images to file.

The Diagram Menu

Action	Usage	Shortcut	See also
Properties	View and edit the <type> Diagram: <name> dialog for the current diagram.	F5	
Find in Project Browser	Locate the current diagram in the Project Browser window.	Shift + Alt + G	
Layout Diagram	Automatically layout the current diagram (not available for Behavioral diagrams).		Layout a Diagram Automatically ^[787]
Switch to	Change the presentation of the diagram to the Diagram List format or the Gantt Chart format.		Diagram List ^[596]
Advanced	Display the diagram Advanced menu.		Diagram Advanced Menu ^[106]
Save	Save the current position of all diagram elements.	Ctrl + S	
Save as Image	Save the diagram as a bitmap file (.BMP), Graphics Interchange Format file (.GIF) or Windows Metafile (.WMF).	Ctrl + T	
Copy Image to Clipboard	Copy an image of the current diagram to the clipboard. The image can be in metafile or bitmap format; you set the format on the Options dialog.	Ctrl + B	General ^[544] (page of Options dialog)
Swimlanes and Matrix	Add, modify and delete swimlanes or the swimlanes matrix for the current diagram.		Swimlanes ^[751] Swimlanes Matrix ^[753]
Repeat Last Element	Create an instance of the same type as the last element created.	Shift + F3	

Action	Usage	Shortcut	See also
Repeat Last Connector	Create an instance of the same type as the last connector created.	F3	
Property Note	Display the properties note for the current diagram on screen.		Insert Diagram Properties Note ^[747]
Show Grid	Show or hide the diagram grid.		
Snap To Grid	<p>Position elements on the diagram grid; there are two options:</p> <ul style="list-style-type: none"> • Standard Grid - constrains elements to the grid when they are added to diagrams • Smart Placement - places elements even distances away from other elements and spaces elements evenly <p>If neither of these options are enabled, the elements can be placed freely on the diagram.</p>		
Zoom	Change the zoom factor on the current diagram.		Pan and Zoom a Diagram ^[763]

2.5.3.5.1 Diagram Advanced Menu

The Diagram **Advanced** menu enables you to perform a number of very useful operations, such as hiding connectors on the diagram or creating a UML Pattern from the diagram.

The Diagram Advanced Menu

Action	Usage	Shortcut	See also
Lock Diagram	<p>Prevent the diagram from being edited, or release the locked diagram for editing.</p> <p>This does not apply in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions, if security is enabled.</p>		Lock Model Elements ^[271]
Make Model Default	Make the current diagram the default diagram opened when the current model is re-opened (unless you set a User Default diagram, which overrides the model default; see below).		Set the Default Diagram ^[744] Project Shortcuts ^[183]

Action	Usage	Shortcut	See also
	<p>This option is also overridden by a project shortcut, which defines displays to present on opening the project.</p> <p>To cancel a Model Default diagram, either:</p> <ul style="list-style-type: none"> • Create a dummy diagram, set it as the Model Default and delete it, or • Delete the original diagram (if it is no longer relevant) 		
Make User Default	<p>(Use in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled).</p> <p>Make the current diagram the default diagram opened when you re-open this model.</p> <p>The User Default diagram overrides the Model Default diagram (see Make Model Default, above); this option is itself overridden by a project shortcut, which defines displays to present on opening the project.</p> <p>To cancel a User Default diagram, either:</p> <ul style="list-style-type: none"> • Create a dummy diagram, set it as the User Default and delete it, or • Delete the original diagram (if it is no longer relevant) <p>This still blocks the Model Default diagram, whilst Security is enabled; to re-establish the Model Default diagram, set it as the User Default.</p>		Project Shortcuts ^[183]
Visible Relations	Hide or show individual connectors for the current diagram.	Ctrl + Shift + I	Relationship Visibility ^[953]
Sequence Messages	Change ^[1244] the order of the communication messages in the current diagram.		Message (Communication Diagrams) ^[1242]
Change Type	Change the type of the current diagram.		Change Diagram Type ^[740]
Save UML Pattern	Save the current diagram as a UML pattern.		Create a Pattern ^[1271]

2.5.3.6 Element

The **Element** menu enables you to configure and access element features and details and control element layout, and to generate documentation and work on source code for the element

The Element Menu

Action	Usage	Shortcut	See also
Properties	View the Properties dialog of the selected element.	Alt + Enter	Properties Dialog ^[843]
Linked Document	Link the element to a rich text document.	Ctrl + Alt + D	Link the element ^[929]
Advanced	Display the element Advanced submenu.		Advanced Submenu ^[109]
Find in Project Browser	Find the currently selected element in the Project Browser. If no element is selected, find the current diagram in the Project Browser.	Alt + G	
Find in Diagrams	Display all occurrences of the currently selected element in any diagram.	Ctrl + U	
Attributes	View and edit the attributes for the selected element.	F9	Element attributes ^[887]
Operations	View and edit the operations (methods) for the selected element.	F10	Operations ^[907]
Feature Visibility	Select which features and characteristics of the selected element are visible on a diagram.	Ctrl + Shift + Y	Visible ^[745]
Inline Features	Display the element Inline Features submenu.		Inline Features Submenu ^[110]
Source Code Engineering	Display the element Source Code Engineering submenu.		Source Code Engineering Submenu ^[111]
Open Source in External Editor	Open the source code of the selected Class. Source code must have been generated, and the selected element must be a Class.	F12	Editing Source Code ^[1799]

Action	Usage	Shortcut	See also
Appearance	Display the element Appearance submenu.		Appearance Submenu ^[112]
Alignment	Display the element Position submenus.		Position Submenus ^[112]
Make Same			
Z Order			
Size			
Move			
Space Evenly			

2.5.3.6.1 Advanced Submenu

The **Advanced** sub-menu provides various options to choose from to customize the appearance of model elements. Select the **Element | Advanced** menu option to display this submenu.

The Element Advanced Menu

Action	Usage	Shortcut	See also
Add Tagged Value	Add a Tagged Value to the currently selected element.	Ctrl + Shift + T	Quick Add - Tagged Value To Elements ^[97]
Custom References	Show model element cross references.	Ctrl + J	Set Up Cross References ^[803]
Rich Text Format (RTF) Report	Generate a report for the currently selected package in rich text format.		RTF Documents ^[225]
Set Parents and Interfaces	Manually set the element's parents or an interface that it realizes.	Ctrl + I	Set Element Parent ^[80]
Embedded Elements	Attach elements to the currently selected element.	Ctrl + Shift + B	Embedded Elements Window ^[835]
Overrides &	Automatically override methods from parent	Ctrl + Shift +	Override Parent

Action	Usage	Shortcut	See also
Implementations	Classes and from realized interfaces.	O	Operations ^[908]
Change Type	Change the element type of the selected element.		Change Element Type ^[809]

2.5.3.6.2 Inline Features Submenu

Access **Element | Inline Features**

The Inline Features Menu

Action	Usage	Shortcut	See also
Edit Selected	Type over or delete the text of the highlighted feature.	F2	
View Properties	Open the dialog containing details of the selected element feature, or the element if no feature is selected.		
Insert New After Selected	Insert a new item in the element.		
Create Linked Note	Add a Note element linked to the selected item, reflecting the content of that item.		
Add Attribute	Add an attribute to the current element.	Ctrl + Shift + F9	Add an attribute ^[885]
Add Operation	Add an operation to the element.	Ctrl + Shift + F10	Add an operation ^[900]
Add Other	Insert other features such as Maintenance and Testing items.	Ctrl + F11	
Delete Selected from Model	Delete the selected item from the model.	Ctrl + Shift + Delete	



Active keys in this mode include

Key	Action
Enter	Save current changes.
Ctrl+Enter	Save current changes and open a new slot to add a new item.
Esc	Cancel editing.
Shift+F10	Display the context menu for in-place editing.

2.5.3.6.3 Source Code Engineering Submenu

To forward and reverse engineer code using the language of your choice, select the **Element | Source Code Engineering** menu option.

The Source Code Engineering Menu

Action	Usage	Shortcut	See also
Generate Current Element	Generate source code  for the currently selected element.	F11	Generate source code 
Synchronize Current Element	Synchronize the selected Class with the source code.	F7	
Batch Generate Selected Element (s)	Batch generate source code for the currently selected element(s).	Shift + F11	
Batch Synchronize Selected Element (s)	Batch synchronize the currently selected element(s) with source code.	Ctrl + R	
Open Source Directory	Open the directory containing the source for this element.	Ctrl + Alt + Y	

2.5.3.6.4 Appearance Submenu

The **Appearance** sub-menu provides various options to choose from to customize the appearance of model elements.

The Appearance Menu

Action	Usage	Shortcut	See also
Default Appearance	Set border, font and background color and border thickness for the selected element, as its default appearance.	Ctrl + Shift + E	Default appearance [817]
Autosize	Auto-size a group of elements in a diagram to a best fit.	Alt + Z	
Set Font	Change the font of the text displayed on the element in a diagram.		Text font [839]
Alternate Image	Select an alternative image for the selected element.	Ctrl + Shift + W	
Apply Image From Clipboard	Insert the image currently held on the clipboard.		

2.5.3.6.5 Position Submenus

These Element menu submenus enable you to size and position elements on the diagram, relative to each other.

The Alignment Sub-Menu

Use the **Alignment** sub-menu to align the selected element(s) to each other.

Action	Usage	Shortcut	See also
Left	Align left edges of elements.	Ctrl + Alt + Left	
Right	Align right edges of elements.	Ctrl + Alt + Right	
Top	Align top edges of elements.	Ctrl + Alt + Up	
Bottom	Align bottom edges of elements.	Ctrl + Alt + Down	

Action	Usage	Shortcut	See also
Centers	Align centers of elements, horizontally or vertically.		

The Make Same Sub-Menu

Use the **Make Same** sub-menu to make the selected elements the same width, the same height or both.

In Use Case diagrams, these options are blocked if the **Allow Elongated Use Cases** option is turned off on the Options dialog Objects page.

Action	Usage	Shortcut	See also
Width	Make all selected elements the same width.		Objects ^[556]
Height	Make all selected elements the same height.		
Both	Make all selected elements the same height and width.		

The Z Order Sub-Menu

Use the **Z Order** sub-menu to move the selected element(s) back, forward, to the front of all other elements or behind all other elements.

Action	Usage	Shortcut	See also
Forward	Move selected element forward one position in the Z order.		Z Order Elements ^[744]
Back	Move selected element back one position in the Z order.		
To Bottom	Send selected element behind all other elements.		
To Top	Bring selected element in front of all other elements.		

The Size Sub-Menu

Use the **Size** sub-menu to make the selected element(s) wider, narrower, taller or shorter by one increment.

Action	Usage	Shortcut	See also
Wider	Make all selected elements wider.	Ctrl + →	
Narrower	Make all selected elements narrower.	Ctrl + ←	
Shorter	Make all selected elements shorter.	Ctrl + ↑	
Taller	Make all selected elements taller.	Ctrl + ↓	

The Move Sub-Menu

Use the **Move** sub-menu to move the selected element(s) left, right, up or down by one increment.

Action	Usage	Shortcut	See also
Right	Move all selected elements to the right.	Shift + →	
Left	Move all selected elements to the left.	Shift + ←	
Up	Move all selected elements upwards.	Shift + ↑	
Down	Move all selected elements downwards.	Shift + ↓	

The Space Evenly Sub-Menu

Use the **Space Evenly** sub-menu to distribute the selected elements evenly.

Action	Usage	Shortcut	See also
Across	Space elements evenly, horizontally.	Alt + -	
Down	Space elements evenly, vertically.	Alt + =	

2.5.3.7 Tools

The **Tools** menu provides access to various tools including those related to code engineering, model management, spell checking, customization of features and setting operational options.

The Tools Menu

Action	Usage	Shortcut	See also
Open Source File	Open any type of external source file (code, XML, DDL) for editing.	Ctrl + Alt + O	Compare Editors ^[180]
Source Code Engineering	Display the Source Code Engineering submenu.		Source Code Engineering ^[116]
Database Engineering	Display the Database Engineering submenu.		Database Engineering ^[117]
Web Services	Display the Web Services submenu.		Web Services ^[118]
XML Schema	Display the XML Schema submenu.		XML Schema ^[118]
Model Transformation (MDA)	Display the Model Transformation submenu.		Model Transformation ^[118]
Spelling Tools	Configure and run the spelling checker.		Configure and Run ^[119]
Data Management	Manage your project's data.		Data Management Submenu ^[119]
MDG Technology Import	Import an MDG Technology file to APPDATA. (Not available in the Desktop edition.)		Import ^[128] a file
Generate MDG Technology File	Display the MDG Technology Wizard. (Not available in the Desktop edition.)		MDG Technology Wizard ^[132]
ODBC Data Sources	Open the Windows system ODBC Data sources administration tool. This is useful when defining connections to EA repositories, or when setting up connections to data sources for reverse engineering purposes.		The Customize Dialog ^[153]
System Information	Open the Windows System Information tool which allows you to explore the various		The Customize Dialog ^[153]

Action	Usage	Shortcut	See also
	configurations and settings active on your current system.		
Customize	Customize the operation of Enterprise Architect. Allows you to modify the main menu, toolbars, tools and other user interface behavior to your particular preferences.		The Customize Dialog ^[153]
Options	Customize your general settings through the Options dialog. These options govern how many processes within EA are executed, and how various elements are displayed.	Ctrl + F9	Options dialog ^[544]

2.5.3.7.1 Source Code Engineering Submenu

To forward and reverse engineer code using the language of your choice select the **Tools | Source Code Engineering** menu option (Professional, Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions).

The Source Code Engineering Menu

Action	Usage	Shortcut	See also
Generate Package Source Code	Generate source code for the currently selected package.	Ctrl + Alt + K	Generate source code ^[1772]
Synchronize Package Contents	Synchronize the selected package with the source code.	Ctrl + Alt + M	
Import Source Directory	Import and reverse engineer an entire directory structure.	Ctrl + Shift + U	Import ^[1795] an directory structure
Import Binary Module	Import a binary module.		Import a binary module ^[1797]
Import ActionScript Files	Import code written in ActionScript with the file extension .AS.		ActionScript ^[1793] code
Import C Files	Import code written in ANSI C with the file extension .C or .H.		ANSI C ^[1793] source code

Action	Usage	Shortcut	See also
Import C# Files	Import code written in the C# programming language with the file extension .CS.		C# ^[1794] source code
Import C++ Files	Import code written in the C++ programming language with the file extension .H, .HPP or .HH.		C++ ^[1794] source code
Import Delphi Files	Import code written in the Delphi programming language with the file extension .PAS.		Delphi ^[1794] source code
Import Java Files	Import code written in the Java programming language with the file extension .JAVA.		Java ^[1794] source code
Import PHP Files	Import code written in PHP with the file extensions .INC, .PHP or .PHP4.		PHP ^[1795] source code
Import Python Files	Import code written in Python with the file extension .PY.		Python ^[1795] source code
Import Visual Basic Files	Import code written in the Visual Basic programming language with the file extension .FRM, .CLS, .BAS or .CTL.		Visual Basic ^[1795] source code
Import VB.NET Files	Import code written in the VB.NET programming language with the file extension .VB.		VB.NET ^[1795] source code

2.5.3.7.2 Database Engineering Submenu

Access **Tools | Database Engineering**

The Database Engineering Menu

Action	Usage	Shortcut	See also
Import DB Schema from ODBC	Import a database schema from an ODBC data source.		Import a database schema from an ODBC ^[1973]
Generate Package DDL	Generate a Package DDL script to create the tables in the currently selected package.		Generate a Package DDL ^[1973]

2.5.3.7.3 Web Services Submenu

Access **Tools | Web Services**

The Web Services Menu

Action	Usage	Shortcut	See also
Import WSDL	Reverse engineer a Web Service Definition Language (WSDL) file as a UML Class model.		Reverse engineer [2088]
Generate WSDL	Forward engineer a UML Class model to a WSDL file.		Forward engineer [2085]

2.5.3.7.4 XML Schema Submenu

Access **Tools | XML Schema**

The XML Schema Menu

Action	Usage	Shortcut	See also
Import XML Schema	Reverse engineer a W3C XML Schema (XSD) file as a UML Class model.		Reverse engineer [2053]
Generate XML Schema	Forward engineer a UML Class model to a W3C XML Schema (XSD) file.		Forward engineer [2049]

2.5.3.7.5 Model Transformation (MDA) Submenu

Access **Tools | Transformations**

The Model Transformation Menu

Command	Action	Shortcut	See also
Transform Selected Elements	Perform an MDA-style transformation on the currently selected elements.	Ctrl + H	Selected elements [1683]
Transform Current Package	Perform an MDA-style transformation on the currently selected package.	Ctrl + Shift + H	Selected package [1683]

2.5.3.7.6 Spelling Tools

Access [Tools](#) | [Spelling Tools](#)

The Spelling Tools Menu

Action	Usage	Shortcut	See also
Spell Check Project	Spell check the whole project.	Ctrl + F7	Spell check ^[485]
Spell Check Current Package	Spell check the current package, selected in the Project Browse.	Ctrl + Shift + F7	Spell check ^[485]
Spelling Options	Specify a range of options to configure how the spell checker operates.		Configure options ^[487]

2.5.3.7.7 Data Management Submenu

Use the options on the **Data Management** submenu to manage your project's data.

The Data Management Menu

Action	Usage	Shortcut	See also
Project Transfer	Move a complete project from one repository to another. You cannot move a project from a source .EAP file of a version earlier than 3.5.0.		Perform a Project Data Transfer ^[442]
Reset IDs	Reset the table auto increment column values.		Reset Table Auto Increment or Identity Columns ^[536]
Project Compare	Compare the total project sizes of two projects.		Compare Projects ^[445]
Project Integrity Check	Check the data integrity of a project.	Shift + F9	Check Project Data integrity ^[535]
Run Patch	Execute an SQL Patch.		Execute an SQL Patch ^[539]
Manage .EAP File	Repair, compact or replicate your .EAP file.		Repair, compact or replicate ^[120]

Action	Usage	Shortcut	See also

2.5.3.7.7.1 Manage .EAP File Submenu

Use the options on this submenu to repair, compact or replicate your .EAP file.














The Manage EAP Files Menu

Action	Usage	See also
Repair .EAP File	Repair an Enterprise Architect project. If a project has not been closed properly, in rare cases it might not open correctly; this option attempts to repair such projects. All users must be logged off the project while it is being repaired.	Repair ^[540] a project
Compact .EAP File	Compact an Enterprise Architect project. Eventually projects might benefit from compacting to conserve space. Ensure everyone is logged off the target project, then select this option to compact the project.	Compact ^[539] a project
Make Design Master	Make a design master project; this is the master project for creating replicas.	Make a design master ^[246] project
Create New Replica	Create a new replica from the Design Master.	New replica ^[244] Design Master ^[246]
Synchronize Replicas	Copy changes from one replica set member to another.	Copy changes ^[247]
Remove Replication	Remove all replication features if you no longer require a model to be replicable.	Remove ^[248] all replication features
Resolve Replication Conflicts	Resolve any conflicts caused when multiple users have changed the same element between synchronization points.	Resolve any conflicts ^[250]

2.5.3.8 Analyzer

The **Analyzer** menu enables you to perform build, debug and profiling operations on your model code.

The Analyzer Menu

Action	Usage	Shortcut	See also
Execution Analyzer	Display the Analyzer Scripts window, which enables you to create and export Build Scripts.		Managing Analyzer Scripts 
Build	Build  the application for your current script (selected on the Execution Analyzer window), and execute the Build command in the Build Scripts dialog.	Ctrl + Shift + F12	Build  the application
Debug	Display the Debug  menu, which provides a set of debugging commands.		Debug  menu
Record	Display the Record  menu, which provides various options for recording a debug session.		Record  menu
Profile	Display the Profiler  window, which enables you to report on runtime aspects such as: <ul style="list-style-type: none"> • What functions are most-frequently called in an application • Which tasks are taking longer than expected, and • Which tasks are taking the longest time to execute 		Profiler  window
Test	Display the Test submenu, which provides access to the Testpoint Manager and executing a Test script.		Test Menu 
Simulation	Display the Simulation window, which enables you to simulate the debugging of an application during the design of that application.		
Breakpoints & Events	Display the Breakpoints & Events window, which you use to manage breakpoints and markers for debugging.		Manage Breakpoints and Markers 
Call Stack	View the currently-running threads in a debugging session, on the Call Stack.		Capture Current Work Environment 
Locals	View the local variables and their values for the current local thread, on the Locals window.		Locals  window

Action	Usage	Shortcut	See also
Watches	Evaluate data items that are not available as local variables - such as data items with static Class member items, on the Watches window.		Watches ^[1880] window
More Analyzer Windows	Display the More Analyzer Windows submenu, which provides access to the Package Build Scripts dialog, the Workbench window, the Modules window and the Memory Viewer window.		More Analyzer Windows ^[125]
Run	Execute the Run command you configured in the Build Scripts dialog.	Ctrl + Alt + N	Add Run Command ^[1862]
Deploy	Execute the Deploy command you configured in the Build Scripts dialog.	Ctrl + Shift + Alt + F12	Deploy ^[1862] command

2.5.3.8.1 Debug Menu

The Debug menu provides a range of options for running a debug session.

The Debug Menu

Action	Usage	Shortcut	See also
Debugger	Display the Debugger ^[150] window, which enables you to control debugging and shows output from a debug session.	Alt + 8	
Debug Run	Start or resume the debugger. If no debug session is in progress, the active Analyzer Script is used to determine which application to run or whether the debugger should attach to an already running process.	F6	Run the Debugger ^[1875]
Debug Pause	Suspend execution of process being debugged.		
Step Into	Step into the current function.	Shift + F6	
Step Over	Step over the current function.	Alt + F6	
Step Out	Step out of the current function.	Ctrl + F6	

Action	Usage	Shortcut	See also
Debug Stop	Stop the current debug session.	Ctrl + Alt + F6	
Generate Sequence Diagram From Stack	Create a Sequence diagram from the Stack Trace History.		Create Sequence Diagram Of Call Stack ^[1886]
Register as Just-In-Time Debugger	Registers Enterprise Architect as the operating system debugger, saving the currently registered debugger. When an exception or crash occurs in an application running outside Enterprise Architect, the Enterprise Architect debugger is opened to enable you to debug the application. When the option is de-selected again, the previously-registered debugger is restored.		

2.5.3.8.2 Record Menu

The **Record** menu provides various options for recording a debug session.

The Record Menu

Action	Usage	Shortcut	See also
Recorder	Display the Record & Analyze window, which enables you to review the information recorded during a debug session.		The Recording History ^[2149] Recorder Toolbar ^[2163]
Start Debug Recording	Enabled when the program is at a breakpoint. Enter manual record mode; each time you step into a function, the call is captured. When you step out, the return is captured.		
Stop Debug Recording	Enabled when recording is in progress. Stops recording.		

Action	Usage	Shortcut	See also
Auto Record Thread	<p>Enabled when the program is at a breakpoint.</p> <p>The thread that is halted is recorded, and the program resumes execution; whenever the thread makes a function call or a function call exits, this information is captured to the History window.</p> <p>The Stack Trace History, Stack tab and Source Code Editor dynamically update to reflect the current execution sequence for the thread; Stack Trace Recording ends when the thread ends or when you click on the Stop button.</p>		
Show/Hide Execution	<p>Display the executing code while the recording is in progress; this slows recording down considerably, and is not used when recording is being controlled by recording markers.</p> <p>The Call Stack is updated in realtime, even while recording is in progress.</p> <p>Watching this window while recording can give a quick picture of what is going on, even without producing a Sequence diagram.</p>		View the Call Stack ^[1884]
Generate Sequence Diagram from Recording History	Create a Sequence diagram from the Record & Analyse window.		Generating Sequence Diagrams ^[2164]

2.5.3.8.3 Test Menu

The Test submenu provides access to the Testpoint Manager and enables you to run test scripts.

The Test Menu

Action	Usage	Shortcut	See also
Testpoint Manager	Display the Testpoints window to execute test scripts and monitor or control the output.		The Testpoints Window ^[2189]
Run Test Script	Execute the Test command you configured in the Build Scripts dialog.	Ctrl + Alt + T	Add Testing Command ^[1834]

2.5.3.8.4 More Analyzer Windows

The **More Analyzer Windows** menu provides access to additional analysis tools.

The More Analyzer Windows Menu

Command	Action	Shortcut	See also
Analyzer Scripts	Create and configure Package Build compiler scripts.	Shift + F12	Managing Analyzer scripts ^[1828]
Workbench	Display the Workbench window, which enables you to create your own workbench variables and invoke methods on them.		Object Workbench ^[2176] Create workbench variables ^[2177] Invoke methods ^[2179]
Modules	Review details of the loaded modules, on the Modules window.		Show Loaded Modules ^[1887]
Memory Viewer	Review a process memory debugging session, using the Memory Viewer window.		Inspect Process Memory ^[1886]

2.5.3.9 Extensions

The **Extensions** menu includes both dynamic entries for Add-ins (extensions written in COM that interact with the Enterprise Architect user interface and models) and standard entries for Importing, Publishing and working with some specific technologies.

Standard Enterprise Architect Based Menu Items

Action	Usage	Shortcut	See also
ArcGIS	This menu is for the built in MDG Technology for ArcGIS, a technology supporting geospatial data modeling. From this menu you can create new model packages using patterns, import ArcGIS data models, export models to schema and otherwise configure the ArcGIS technology.		ArcGIS UML Profile ^[1633]
ODM	This menu is for the built in MDG Technology for Ontology Definition Metamodel (ODM), a technology for developing large-scale ontologies for project domains. From this menu you can import OWL/RDF XML files, export packages as OWL/RDF XML files, define new namespaces for ODM packages and define new labels for OWL and RDF elements.		MDG Technology for ODM ^[1643]

Action	Usage	Shortcut	See also
Import	This menu option supports importing XMI/XML from a variety of formats. ArcGIS, OWL/RDF and EMX are specifically supported, but a wide range of other tools' XML output can be imported into Enterprise Architect using the Other Tools/Formats menu option.		ODM Commands ^[1652] Import ArcGIS XML Workspace ^[1641]
Publish	<p>Publishing is the export of Enterprise Architect model information in a specific format for consumption by other tools or processes. A wide range of output formats is supported - and you can further tailor the output by including or excluding diagram information, Enterprise Architect specific extensions and so on. You should tailor your published file for the capabilities of the tool or process expected to consume it.</p> <p>The Normative XMI option provides a quick means of exporting model information in the latest UML/XMI combination with no Enterprise Architect extension information or diagram information. This format is most useful for standards development and for base interoperability with other UML based tools.</p> <p>Note that you should not use this menu to Publish model information for consumption by other users of Enterprise Architect. For that purpose, use the Project Model Import/Export functions.</p>		Publish Model Package ^[414] Import/Export Submenu ^[102] ODM Commands ^[1652] Export ArcGIS XML Workspace ^[1640]

Dynamic and Configuration Menu Items created and/or used when Add-Ins such as TOGAF, UPDM and others are installed

Action	Usage	Shortcut	See also
<Add-In Name(s)>	<p>Access the facilities of the selected Add-In, through an Add-In submenu (below).</p> <p>For example, if you have TOGAF enabled on your system, you might click on the TOGAF menu option and display a submenu of options available to the TOGAF user in Enterprise Architect.</p>		
Add-In Windows	<p>Display a window, or list of windows, provided by any Add-Ins you have installed and enabled.</p> <p>If no windows are provided, displays an empty, docked Add-Ins window.</p>		

Action	Usage	Shortcut	See also
Manage Add-Ins	Display the Manage Add-Ins [530] dialog, which you use to enable or disable Add-Ins for use. Any technology loaded by an Add-In is automatically enabled; if you do not want to use it, you can disable it on the dialog.		Manage Add-Ins [2593] dialog

Add-In Submenu

Action	Usage	Shortcut	See also
<Add-In specific options>	List options to perform various functions specific to the Add-In. For example, the MDG Technology For Zachman Framework, as an Add-In, has the options Open Example Model and Insert New Framework Model .		
Help	Display the Help subsystem for the Add-In.		
About	Display information on the Add-In installation, such as version, registration details and copyright statement.		

2.5.3.10 Settings

The **Settings** menu enables you to configure various settings for your overall project, such as stereotypes, Tagged Values, cardinality values, datatypes, language macros, local directories and image management.

The Settings Menu

Action	Usage	Shortcut	See also
UML Types	Configure stereotypes, Tagged Value Types and the cardinality list for your project.		Stereotypes [982] Tagged Value Types [985] Cardinality list [986]
Project Types	Display the Project Types submenu.		Project Types [129]
Namespace Roots	Locate and delete model namespaces.		Model namespaces [177]

Action	Usage	Shortcut	See also
Project Template Package	Configure or change the default element template directory.		
Auto Names and Counters	Configure automatic naming and counters for elements.		Use Auto Naming and Auto Counters ^[799]
MDG Technologies	Display the MDG Technologies dialog, which enables you to activate and use MDG Technology files.		MDG Technologies ^[1285]
Document Template Designer	Displays the Document Template Designer for creating and reviewing RTF document templates.		Document Template Designer ^[2272]
Code Generation Templates	Modify code generation templates using the Code Templates Editor.	Ctrl + Shift + P	Code Templates Editor ^[1924]
Local Directories and Paths	Configure local directories and paths.		Local directories and paths ^[1900]
Code Datatypes	Add, modify and delete programming languages datatypes.		Programming languages datatypes ^[987]
Preprocessor Macros	Add and delete preprocessor macros.		Preprocessor macros ^[1902]
Database Datatypes	Add, modify and delete database datatypes.		Database datatypes ^[1947]
Transformation Templates	Modify transformation templates using the Transformation Templates Editor.	Ctrl + Alt + H	Transformation Templates Editor ^[1710]
Images	Open the Image Manager.		Image Manager ^[755]
Colors	Configure the custom colors for the project. There are two options: <ul style="list-style-type: none"> • Get Project Custom Colors - get the custom colors • Set Project Custom Colors - set the custom colors 		Appearance ^[817] dialog

Action	Usage	Shortcut	See also
	Custom colors are as used in the Appearance dialog.		

2.5.3.10.1 Project Types Submenu

The **Project Types** submenu enables you to create and manage project variables and definitions for use across the project.

The Project Types Menu

Action	Usage	Shortcut	See also
People	Display the People dialog, which enables you to configure the authors, clients, resources and roles for your project.		People ^[989] dialog
General Types	Display the General Types dialog, which enables you to configure requirements, status types, constraints and scenarios for your project.		General Types ^[995] dialog
Project Indicators	Define the project indicators (risks, efforts and metrics) used in Resource Management.		Risks ^[458] Efforts ^[456] Metrics ^[457]
Maintenance	Display the Maintenance dialog, which enables you to track problem types and test types.		Maintenance ^[1007]
Estimation Factors	Display the Estimation factors dialog, which enables you to configure estimation factor types (Technical Complexity Factors, Environmental Complexity Factors, and Default Hour Rate) for your project.		Estimation ^[516] factor Technical Complexity Factors ^[517] Environmental Complexity Factors ^[518] Default Hour Rate ^[520]

2.5.3.11 Window

The **Window** menu provides access to various actions related to configuring open windows.

The Window Menu

Action	Usage	Shortcut	See also
Full Screen	<p>Reset the display to full screen so that only the work area and main menu are shown - no toolbars or windows.</p> <p>To return to your normal working display, either click on the Full Screen option again or click on the Close Full Screen pop-up menu option.</p> <p>You can also restore individual menus and toolbars using the View menu options.</p>		
Save All Modified	Save all modified data.		
Reload Current View	Refresh the current view.		Refresh the current view ^[242]
Set Focus to Current View	Make the current view the active one, so that key strokes immediately act on that view.	Ctrl + Shift + 0	
Close Current View	Close the current view.		
Close All Except Current	Close all except the currently selected view.		
Close All	Close all opened windows in the main tab view.		
Autohide Active Window	Autohide ^[137] the window that currently has focus.	Ctrl + Shift + F4	Autohide ^[137] windows
Autohide All Docked Windows	Autohide ^[137] all windows that are docked.		Autohide ^[137] windows
Close Active Window	Close the window that currently has focus.	Ctrl + F4	
Show Start Page	<p>Show or hide the Start Page.</p> <p>The setting of this option persists through shut down and restart, until you specifically change the option again.</p>		Start Page ^[87]
Always on Top	Force the main Enterprise Architect window to be on top of all other windows.		

2.5.3.12 Help

The **Help** menu provides access to the Enterprise Architect help files, the Read Me file, the Enterprise Architect End User License Agreement and various features on the [Sparx Systems website](#).

The Help Menu

Action	Usage	Shortcut	See also
About EA	View information about Enterprise Architect, including your registration details.		
Help Contents	Display the Enterprise Architect Help, starting at the introductory page.		
On-Line Resources	Display the Online Resources submenu; see below.		
Open Example Model	Open the <i>EA.Example</i> model provided with the Enterprise Architect installer.		
Register and Manage License Key(s)	Configure and manage the license keys used to register the name and keys for Enterprise Architect and its Add-Ins.		License Management [2736]
View License Agreement	View the Enterprise Architect End User License Agreement.		
Ordering Information	View information on how to purchase Enterprise Architect.		Order Enterprise Architect [35]
Read Me	View the <i>Readme.txt</i> file, which details the changes and enhancements in Enterprise Architect, build by build.		
Keyboard Accelerator Map	View the keyboard accelerator map. You can customize your keyboard shortcuts, if required.		Customize your keyboard shortcuts [160]
EA on the Web	Visit the Sparx Systems website.		Sparx Systems website

The On-Line Resources Sub-Menu

Access help and resources on-line at the Sparx Systems website.

Action	Usage	Shortcut	See also
User Forum and News	Visit the Enterprise Architect user forum.		User forum
Request-a-Feature	Request a feature you would like to see in Enterprise Architect.		Request a feature
Bug Report Page	Report the details of a bug you have found in Enterprise Architect.		Report the details of a bug
Latest Version Details	Display the Announcements folder of the user forum, providing details of the latest Enterprise Architect build.		Details of the latest Enterprise Architect build
Automation Interface	Access the Enterprise Architect Automation Interface pages on the Sparx Systems website.		Automation Interface
Introducing UML	Access the Sparx Systems online UML tutorials.		UML tutorials
Pricing and Purchase Options	Purchase or upgrade Enterprise Architect over the internet.		Purchase or upgrade Enterprise Architect

2.5.4 Standard Windows

There are many dockable windows available to use in Enterprise Architect. These can be accessed either:

- Through the **View** menu, or
- Through the context menu accessed by right-clicking on the main menu

Having displayed a window, you can also reduce it to a tab off to one side, or autohide it.

The dockable windows available include

Topic	Link
Project Browser	Project Browser ^[567]
Properties	Properties ^[878]
Project Information	The Project Information Window ^[460]
Testing	Testing ^[2213]

Topic	Link
Maintenance	Maintenance ^[2237]
Toolbox	Toolbox ^[698]
Resources	Resources ^[1003]
Source Code Viewer	Editing Source Code ^[1799]
Scripting window	Scripting window ^[2372]
Debug	Debug ^[1874]
Learning Center	Learning Center ^[89]
Notes	Notes ^[977]
Traceability	Traceability ^[637]
Tagged Values	Tagged Values ^[969]
Project Management	Project Management ^[448]
Model Views	Model Views ^[598]
Element Browser	Element Browser ^[876]
Relationships	Relationships ^[648]
Scenarios & Requirements	Scenarios & Requirements ^[879]
Pan & Zoom	Pan & Zoom ^[610]
Layout Tools	Layout Tools ^[769]
Team Review	Team Review ^[279]

Topic	Link
Diagram Filters.	Diagram Filters ^[629]

Notes

- On the Testing, Maintenance and Project Management windows, any descriptive, history, input or results text for a selected item is also displayed in the Notes window; you cannot edit this text in the Notes window

Learn more

- [Code Editors](#) ^[134]
- [Docking Windows](#) ^[135]
- [Autohide Windows](#) ^[137]

2.5.4.1 Code Editors

Enterprise Architect provides a number of editors that you can use across the development environment to maintain scripts, code and templates. Each editor has its own features, but they are all based on a common *Code Editor control*.

Code Editors

Topic	Detail	See also
Editors	<p>The provided editors are:</p> <ul style="list-style-type: none"> • Shape Script Editor • Code Generation Template Editor • Transformation Template Editor • Custom SQL Search Filter Editor • Database View Editor • HTML Report Template Editor • Source Code Viewer • Script Editor <p>You can have several code editors (or files within a code editor) open at the same time, as separate tabs in the central view area of the Enterprise Architect work area.</p> <p>You can also close the editors individually or all together, leaving views of other types (such as diagrams or RTF reports) still open.</p>	<p>Shape Script Editor ^[1359]</p> <p>Code Generation Template Editor ^[1404]</p> <p>Transformation Template Editor ^[1710]</p> <p>Custom SQL Search Filter Editor ^[623]</p> <p>Database View Editor ^[1968]</p> <p>HTML Report Template Editor ^[2355]</p> <p>Editing Source Code ^[1799]</p> <p>Script Editor ^[2378]</p> <p>Diagram Tabs ^[695]</p>

2.5.4.2 Dock Windows

A number of Enterprise Architect windows can be freely positioned on the screen, or docked against any edge of the *application workspace*. These windows are collectively called dockable windows.

Drag the window around the application workspace until you find a comfortable way of working. The examples below describe a few ways you can rearrange the windows to suit your work habits.


Floating Windows

To float a window anywhere on the screen, just drag the window by its title bar to the required position.

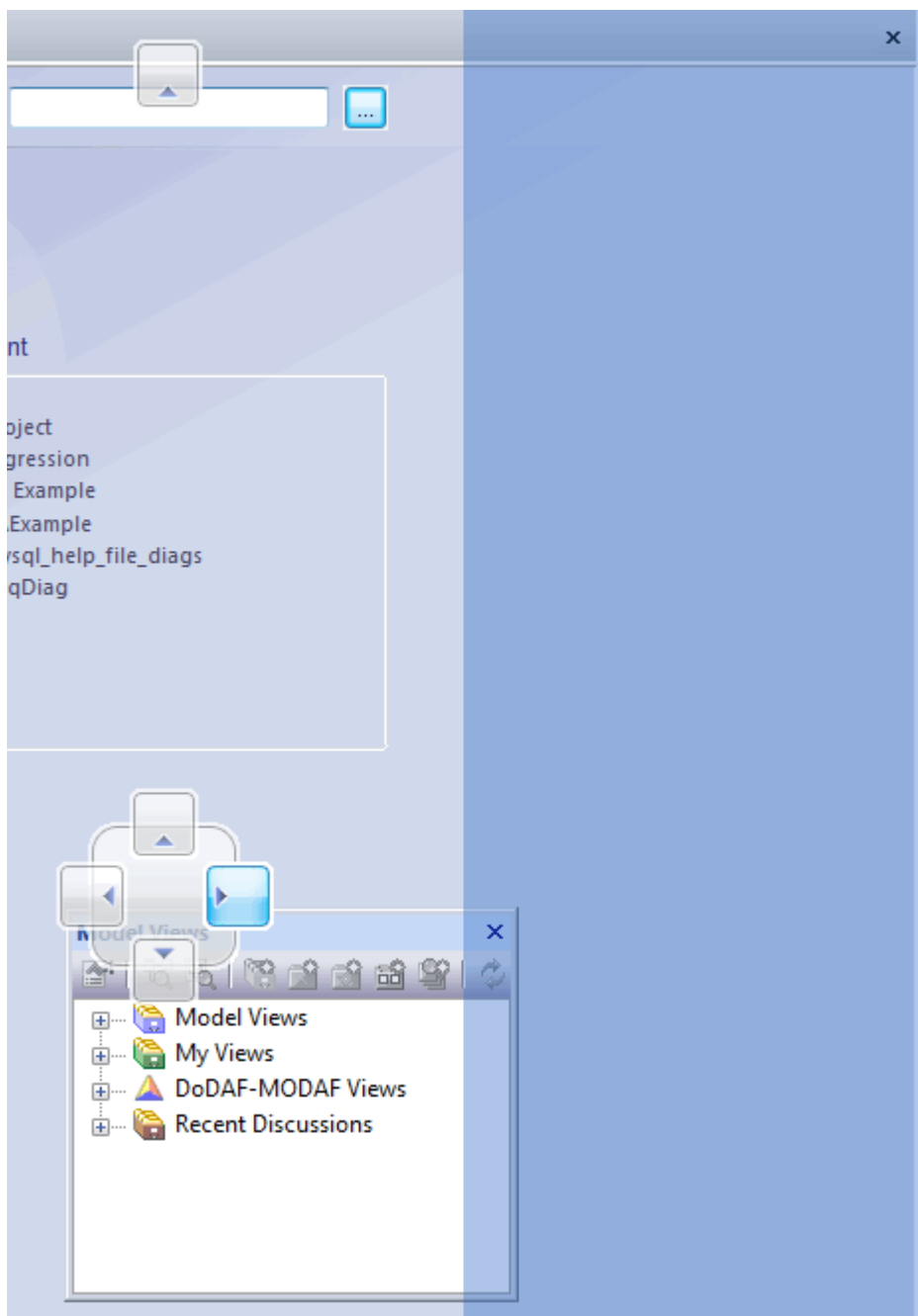
Dock a Window Against an Edge

The *navigation compass* enables you to dock windows against an edge of the application workspace. You drag the window over one of the points of the compass to dock it into a tabbed location. The window does not overlap any other window, so if you are docking several windows you could cover the workspace; however, you can avoid this by combining them in a single tabbed frame.

To dock a window against an edge

Step	Action	See also
1	<p>Click on the item to move and start dragging it towards the required position; this activates the navigation compass.</p> 	
2	<p>Drag the window onto a compass point.</p> <p>The screen display shades the area where the window is to be placed.</p>	
3	<p>Release the mouse button over the compass point to confirm the position; this docks the window.</p>	

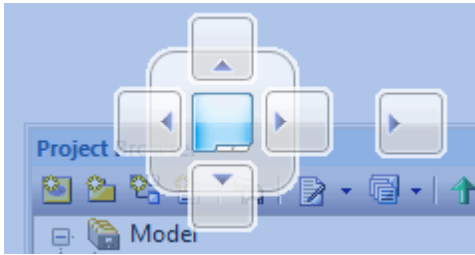
In the example below, when the mouse button is released the Model Views window is docked into the shaded area.



Dock Required Windows into One Frame

You can also dock all of the windows you are using into a single frame, by either:

- Dragging the title bar of each window up to the title bar of the first docked window, or
- Dragging each window over the 'tabbed frame' icon in the middle of the compass, when another window is already docked.



You can do this with all dockable windows.



To separate a window from a combined frame, click on the window tab at the bottom of the frame and drag it away.

Learn more

- [Dockable Windows](#) ^[132]

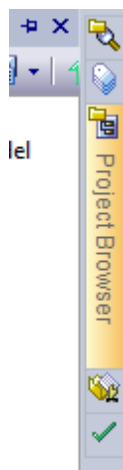
2.5.4.3 Autohide Windows

Autohide Using the Toggle Button

- You can automatically hide browser frames by clicking on the  button, located in the top right corner of the frame
- To turn off the autohide for a particular set of windows within a frame, click on the  button

Use Automatically Hidden Windows

When you automatically hide a set of windows in a frame, the tabs contract to the outside of the application workspace.



Hover the cursor over a window symbol to expand both the tab and the associated window.

You can also use the **View | Visual Style | Animate Autohide Windows** menu option to animate windows that have been automatically hidden.

2.5.5 Standard Toolbars


Enterprise Architect provides a set of toolbars containing convenient shortcuts to a wide range of common functions that you can perform on your project as a whole through to the individual modeling units. You can also customize the toolbars by deleting and reordering the default button set.

The toolbars are initially displayed in the toolbar banner at the top of the workspace, but you can display them at the foot of the workspace, drag and dock them within the workspace, or float them over the application; this is useful when you are using a certain set of functions a lot in a particular area.

You can also dock toolbars to the edge of the Enterprise Architect workspace by dragging them by the title bar and placing them against the appropriate edge.

The toolbars available

Topic	Link
Default Tools Toolbar	Default Tools Toolbar ^[139]
Project Toolbar	Project Toolbar ^[141]
Code Generation Toolbar	Code Generation Toolbar ^[142]
UML Elements Toolbar	UML Elements Toolbar ^[143]
Diagram Toolbar	Diagram Toolbar ^[145]
Current Element Toolbar	Current Element Toolbar ^[146]
Current Connector Toolbar	Current Connector Toolbar ^[148]
Format Toolbar	Format Toolbar ^[692]
Workspace Layouts Toolbar	Workspace Layouts Toolbar ^[149]
Status Bar	Status Bar ^[152]
Notes Toolbar	Notes Toolbar ^[979]

Each toolbar has a drop-down arrow at the right-hand end, , which can be enabled or hidden from the Visual Style Sub-Menu. If you click on this drop-down arrow, the **Add or Remove Buttons** option displays. Select this option to show a context menu listing the toolbars that are currently displayed, and an option to

customize both your own toolbars and the system-provided toolbars.

You can select one of the toolbars identified on the context menu to list the icons available through that toolbar. Click on the icons as necessary to hide or show them in the toolbar.

Learn more

- [Customize Commands](#)^[154]
- [The Visual Style Sub-menu](#)^[99]
- [The Customize Window](#)^[153]

2.5.5.1 Default Tools Toolbar

The Default Tools toolbar provides instant access to the most commonly used tools in Enterprise Architect including saving, printing and technology selection.

Access [View | Toolbars | Default Tools](#)

Use to

- Create new projects and open existing projects
- Save changes to the current diagram
- Edit, cut, copy and paste diagram objects
- Print preview, configure print settings and print the current diagram
- Open the Package Browser
- Define, manage and run Model Searches
- Select the working Technology for toolbox and element creation
- Access Enterprise Architect Help

Reference



Action	Usage	Shortcut	See also
New Project	Create a new Enterprise Architect project (as a .EAP file).	Ctrl + N	New Enterprise Architect Project ^[189]
Open Project	Open an existing project.	Ctrl + O	Open a project ^[187]
Save Diagram	Save the current diagram.	Ctrl + S	

Cut	Cut the selected element(s) from the diagram and copy them to the Enterprise Architect clipboard.	Ctrl + X	
Copy	Copy the selected element(s) from the diagram to the Enterprise Architect clipboard.	Ctrl + Space (Ctrl + C)	
Paste	Paste from the Enterprise Architect clipboard as a reference on the current diagram.	Shift + Insert (Ctrl + V)	
Undo	Undo the last action.	Ctrl + Z	Undo last action ^[768]
Redo	Redo the last undone action.	Ctrl + Y	Redo last action ^[768]
Print Preview	Display the current diagram as it will appear when printed.		
Page Setup	Configure print settings for the current diagram.		
Print	Print the current diagram.	Ctrl + P	
Package Browser	Display the current package as a list of editable elements.		Package Browser ^[587]
Open Model Search	Display the Model Search window, to define, manage and run model searches.	Ctrl + Alt + A	Use model search ^[617]
Technology List	Set the active technology for the Toolbox and the diagram space bar context element menu.		Manage MDG Technologies ^[1285]
Help Contents	Access the Enterprise Architect Help.	F1	Help Menu ^[137]

Notes

- The **Cut** and **Copy** buttons are only enabled when an item is selected on the current diagram
- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Default Tools** menu option

2.5.5.2 Project Toolbar

The Project toolbar provides tools for performing routine tasks such as creating packages, diagrams and elements, searching the model and generating documentation.

Access [View | Toolbars | Project](#)

Use to

- Reload current project
- Create new diagrams, packages and elements
- Perform model searches
- Generate documentation
- Examine project issues
- Review glossary terms
- Set project options

Reference



Action	Usage	Shortcut	See also
Reload Project	Reload the current project.	Ctrl + Shift + F11	Reload current project ^[242]
New Diagram	Create a new diagram in the currently selected package.	Ctrl + Insert	
New Package	Create a new package within the currently selected package.	Ctrl + W	
New Element	Create a new model element under the currently selected package or element.	Ctrl + M	
Search Browser	Run a simple search within the Project Browser.	Ctrl + Shift + F	
Search Model	Search the entire project using pre-defined and customized searches.	Ctrl + F	Model Search ^[619]
RTF Reports	Define and generate RTF project reports.	F8	

Project Issues	Review project issues.		
Project Glossary	Define glossary terms.		
Set Options	Configure display and behaviour options.		

Notes

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Project** menu option

2.5.5.3 Code Generation Toolbar

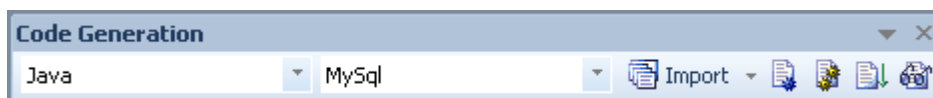
The Code Generation toolbar can import, generate, synchronize and open source code. Convenient menus are provided to set the default language and default database type for each Project.

Access **View | Toolbars | Code Generation**

Use to

- Set the default language
- Set the default database type
- Import Classes and Interfaces from source files
- Generate code for a single selected Class
- Batch generate code for one or more selected Classes
- Synchronize selected Classes with source code
- View code in default editor

Reference



Action	Usage	Shortcut	See also
Set Default Language	Set the language for all newly created elements.		
Set Default Database Type	Set the database type for all newly created database elements.		Physical Data Model [1932]

Import	<p>Choose files to import for reverse engineering source code into the current diagram.</p> <p>Use the drop-down menu to override the project's default language for a particular import.</p> 		
Generate Source Code	Generate Source code for the currently selected element.	F11	
Batch Generate Source Code	Generate Source code for multiple selected elements.	Shift + F11	
Synchronize Element with Source Code	Synchronize the selected element with its associated source code.	F7	
Open Source Code	<p>Open the source code of the selected Class.</p> <p>The source code must have been generated, and the selected element must be a Class.</p>	F12	Editing Source Code ^[1793]

Notes

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Code Generation** menu option

2.5.5.4 UML Elements Toolbar

The UML Elements toolbar can quickly and easily insert UML elements onto a diagram, including Notes, text, hyperlinks and diagram specific UML elements.

Access **View | Toolbars | UML Elements**

Use to

- Insert a variety of elements onto the current diagram

Reference

Action	Usage	Shortcut	See also
New Element	Add a new element to the current diagram. This option displays a list of elements matching the content of the current Toolbox pages, with an Other option to list other elements.	Space bar	Creating Objects ^[797]
New Boundary	Add a simple boundary element to the current diagram.		System Boundary ^[1170]
New Note	Add a new Note element to the current diagram.		Note ^[814]
New Text Element	Add a simple text element to the current diagram.		Text element ^[814]
Diagram Note Element	Add a diagram properties note to the current diagram.		Diagram note ^[747]
Diagram Legend	Add a diagram legend element to the current diagram.		Diagram Legend ^[748]
New Hyperlink	Add a hyperlink element (web page, file, help topic) to the current diagram.		Hyperlink ^[1668]
New Note Link	Create a note link connection from an element to a note in the current diagram.		Note link ^[1249]

Notes

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions

2.5.5.5 Diagram Toolbar

The Diagram toolbar provides tools to lay out and improve the appearance of diagrams, including zoom controls to better manage large diagrams and quickly identify key areas of interest. The Diagram toolbar can be docked to any main window edge or floated freely in a convenient location for quick access.

Access [View | Toolbars | Diagram Tools](#)

Use to

- Layout diagrams
- Set element Z-Order
- Navigate between current open diagrams
- Zoom in and out
- Access properties and delete elements

Reference



Action	Usage	Shortcut	See also
Align Left	Align the selected elements to the left.	Ctrl + Alt + ←	
Align Right	Align the selected elements to the right.	Ctrl + Alt + →	
Align Top	Align the selected elements to the top.	Ctrl + Alt + Up Arrow	
Align Bottom	Align the selected elements to the bottom.	Ctrl + Alt + ↓	
Z Order Top	Bring the selected element to the top of the Z order.		Diagram Context Menu ^[687]
Z Order Bottom	Move the selected element to the bottom of the Z order.		
Previous diagram	Go to the previous diagram.	Alt + ←	
Next Diagram	Go to the next diagram.	Alt + →	
Default Diagram	Go to the default diagram.		

Zoom In	Zoom into the diagram.		
Zoom Out	Zoom out of the diagram.		
Zoom to fit diagram	Zoom to fit the diagram.		
Zoom to fit page	Zoom to fit the page.		
Zoom to 100%	Zoom to 100% of display size.		
Diagram Auto-layout	Auto-layout the diagram (not for Behavioral diagrams) in the Digraph layout.		Digraph ^[778] layout
Diagram Properties	Show the diagram properties.	F5	
Paste Appearance	Paste the diagram appearance as copied into the Painter from an element's Appearance context menu.		Appearance ^[732] context menu
Delete	Delete the selected element(s).	Ctrl + D	

Notes

- Any actions that result in a change in diagram content and appearance (including Zoom) are saved as changes to the diagram
- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Diagram** menu option

2.5.5.6 Current Element Toolbar

The Current Element toolbar performs common operations on a selected element such as viewing properties, operations, attributes. It is also capable of specifying the visibility of element features and compartments and locking an element.

Access **View | Toolbars | Current Element**

Use to


- View and modify element properties
- Set an element's parent or implement interfaces

- View and modify Operations
- View and modify Attributes
- Specify the visibility of element features and compartments
- Specify the run state of an element (or, for Parts, property value)
- View use of element in other structures such as diagrams
- Locate the element in the Project Browser window
- View the cross reference list for this element
- Lock or unlock the current element
- Add a Tagged Value to the current element

Reference



Action	Usage	Shortcut	See also
Edit Properties	View and edit properties of the selected element.	Alt + Enter	
Set Parent	Set the current element's parent or implement interfaces.	Ctrl + I	
Operations	Create and edit operations for the selected element.	F10	
Attributes	Create and edit attributes for the selected element.	F9	
Show Features	Specify which compartments and information to display for the selected element in the current diagram.	Ctrl + Shift + Y	
Set Run State	Set the element run state or attribute initializers.	Ctrl + Shift + R	
Find Element Usage	Find all diagrams in which the selected element is used.	Ctrl + U	
Find in Project Browser	Find the selected element in the Project Browser.	Alt + G	

Configure References	Set and view custom references to other elements and diagrams.	Ctrl + J	
Lock Element	Apply or release a lock on the selected element.		Lock Model Elements 
Add Tagged Value	Add a Tagged Value to the selected element.	Ctrl + Shift + T	

Notes

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Current Element** menu option

2.5.5.7 Current Connector Toolbar

The Current Connector toolbar can easily modify the properties and style of a connector, show and hide labels/connectors and pin each end.

Access **View | Toolbars | Current Connector**

Use to

- View and modify properties for the current connector
- Set the connector line style
- Attach a note to the currently selected connector
- Set the visibility for labels of the connector
- Set the visible or hidden relations in the current diagram
- Reverse the direction of the currently selected connector
- Pin the start and/or connector ends to a position on the target element (drop menu)

Reference



Action	Usage	Shortcut	See also
Edit Properties	View and edit the properties of the selected connector.	Enter	
Set Style	Set the connector line style; choose the style from the list.		

Attach Note	Attach a note or constraint to one or more connectors in the current diagram.		
Show Labels	Set the label visibility for the selected connector.		
Show Relationships	Show and hide specific relationships within the current diagram.	Ctrl + Shift + I	
Reverse Direction	Reverse the direction of the selected connector.		
Pin Connector Ends	Pin the end points of the selected connector.		

Notes

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Current Connector** menu option

2.5.5.8 Workspace Layouts

The Workspace Layouts toolbar provides tools for quickly switching between pre-defined workspace layouts. It also provides tools for laying out your workspace then saving the resulting layout. The Workspace Layouts toolbar can be docked to any main window edge or floated freely in a convenient location for quick access.

Access **View | Toolbars | Workspace Layouts**

Use to

- Save the current workspace layout
- Manage workspace layouts
- Apply a selected workspace layout
- Switch to Views, such as Element List, Model Search and Relationship Matrix
- Activate or select main windows, such as Project Browser, Team Review, Element Properties and Execution Analyzer
- Toggle visibility of toolbars

Reference



Action	Usage	Shortcut	See also
--------	-------	----------	----------

Save As	Save the current workspace layout to a file.		
Manage Layouts	Open the Workspace Layouts dialog.		
Select Layout	Select and apply a saved workspace layout.		
Views	From a drop-down menu, choose one of several views to display: <ul style="list-style-type: none"> • Element list • Model Search • Relationship Matrix • Auditing • Web Browser 		
Windows	Choose, from a drop-down menu, one of many main windows to display or select, such as: <ul style="list-style-type: none"> • Project Browser • Team Review • Element Properties • Execution Analyzer 		
Toolbars	Toggle the visibility of the toolbars.		
Fullscreen	Hide all docked windows and view the current view in full screen.		

Notes

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Workspace Layouts** menu option

2.5.5.9 Debug & Record Toolbar

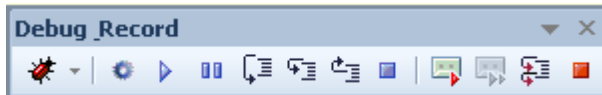
The Debug & Record toolbar provides tools for debugging program source code. The Debug & Record toolbar can be docked to any main window edge or floated freely in a convenient location for quick access.

Access **View | Toolbars | Debug & Record**

Use to

- Attach to running processes
- Run, pause or terminate program execution
- Step over, step into or step out of program statements
- Record program execution

Reference



Action	Usage	Shortcut	See also
Drop-down Command List	Show a drop-down menu containing commands for running and managing Analyzer scripts.		
Attach to Process	Attach to a running process.		
Debug	Begin or continue execution of the current program.	F6	
Pause-Resume	Pause or resume program execution.		
Step Over	Step over the next statement.	Alt + F6	
Step In	Step into the next statement.	Shift + F6	
Step Out	Step out of the current method.	Ctrl + F6	
Debug Stop	Terminate execution of the current program.	Ctrl + Alt + F6	
Manual Record	Manually step record the current thread.		Recording Sequence Diagrams ^[2147]
Auto Record	Auto record an execution trace for the current thread.		
Step Through	Step through the current function. This button is available when manually recording steps; it attempts to step into a function on the line of code. If a function call is made, the function call is	Alt + ;	

	captured and the debugger steps back out of the function to either the same line or the next line.		
Stop Recording	End the recording of program execution.		

Notes

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Diagram** menu option

2.5.5.10 Status Bar

The Status bar displays at the bottom of the Enterprise Architect workspace. It provides feedback on current operations and other status information, and enables you to manipulate the scale of a diagram.

Access **View | Toolbars | Status Bar**

Use to

- Identify the type and name of the currently selected element in a diagram (or the status of a Model Search)
- Identify the name of the currently-selected feature, if one is selected
- View the current coordinates of the top left corner of the selected element, and its width and height
- Adjust the scale of the current diagram by up to 50%, through use of a zoom slider
- Check the status of (**Caps Lock**), (**Num Lock**), (**Scrl Lock**) and the WAN Optimizer (bold indicates 'in use', pale indicates 'off')
- Check, by the presence of a triangle in the bottom right corner, that the screen is not maximized; you can drag the screen corner to increase the size of the window

Reference



Action	Usage	Shortcut	See also
Right-click on Status Bar	Configure the status bar by opening a context menu that enables you to hide or show the element name, element coordinates, zoom slider or status indicators.		
Zoom Out	Zoom out of the current diagram.		

Adjust Zoom	Adjust the scale of the current diagram by moving the slider.		
Zoom In	Zoom in on the current diagram.		

Notes

- This facility has no impact on other users who might view the diagram
- The facility has the same function as the **Scale view by** field on the Diagram Appearance page of the Options dialog; changes in the 'zoomed' display scale of a diagram update this field and are applied to any other diagrams that you open
- The facility also has no impact any other diagram Zoom facility in Enterprise Architect
- You can hide or show the toolbar by clicking on the **View | Toolbars | Status Bar** menu option

Learn more

- [Diagram Appearance](#)^[550]

2.5.6 Customization

You can configure various settings using the Options dialog, which you display by selecting the **Tools | Options** menu option. In addition, there are several options to change the overall look and feel of Enterprise Architect in the **View | Visual Style** submenu. Those settings and options are explored in this topic.

On occasion, you might want to use Enterprise Architect for two distinct types of operation at the same time. You can do this by adding the following command line argument when you run Enterprise Architect:

```
/regkey: <regkeyname>
```

This stores registry settings - such as window layouts - to a different path in the registry.

Learn more

- [Workspace Layouts](#)^[149]
- [Local Options](#)^[542]
- [Visual Styles](#)^[162]

2.5.6.1 The Customize Dialog

The Customize dialog enables you to customize Enterprise Architect:

Access

- Select the **Tools | Customize** option, or
- At the far right of any toolbar, click on the drop-down arrow and on the **Add or Remove buttons** option, then select the **Customize** option

Topic	Link
Commands	Commands ^[154]
Toolbars	Toolbars ^[155]
Tools	Tools ^[156]
Keyboard Keystrokes	Keyboard Keystrokes ^[160]
Menus	Menus ^[161]
Options	Options ^[162]

If a documented toolbar icon, keyboard combination or menu option does not appear to be available, select the appropriate tab and click on the **Reset** or **Reset All** button. This restores the toolbar, menu or key settings to the defaults.

However: Be aware that this also removes any customized icons, options or combinations you might have set, because it is possible that the customization itself has displaced or affected the default setting.

2.5.6.1.1 Customize Commands

The Customize dialog Commands tab provides access to many of Enterprise Architect's functions, enabling you to place them into a toolbar.

To add a command to a toolbar, click on the category in the Categories: panel and select the command from the list for that category in the Commands: panel. Drag the command on top of the toolbar to add it to.

If you right-click on the command icon in the toolbar while the Customize dialog is open, a context-sensitive menu displays. This menu offers options for deleting commands from a toolbar, and for changing the appearance of commands.

To remove a command from the toolbar, right-click on the command graphic or text and select the **Delete** menu option.

To change the appearance of a command graphic, right-click on the command graphic or text and select the **Button Appearance** context menu option. The Button Appearance dialog displays, which you can use to add graphical icons to commands that do not have them by default.

Some commands do not come with a convenient icon, which results in an empty toolbar button. Either avoid placing these commands on toolbars or use the context-sensitive menu to select an appropriate icon for the command.

Read the [Create a New Toolbar and Populate it with Commands](#) ^[155] procedure of the *Customize Toolbars* topic.

2.5.6.1.2 Customize Toolbars



The Toolbars tab on the Customize dialog enables you to:

- Hide or show toolbars by selecting the appropriate checkbox
- Rename toolbars
- Create new toolbars
- Delete toolbars
- Modify toolbar contents by dragging commands onto a visible toolbar from the Commands tab or from another toolbar
- Reset a toolbar (or all toolbars) to the default contents and position, and
- Display text labels under the toolbar icons (perhaps temporarily, just to check what the icons do)

Create a New Toolbar and Populate it with Commands:

To create a new toolbar and populate it with commands

Step	Action	See also
1	Select the Tools Customize menu option. The Customize dialog displays.	
2	Click on the Toolbars tab.	
3	Click on the New button. The Toolbar Name dialog displays.	
4	In the Toolbar Name field, type a name for your new toolbar and click on the OK button. Your new toolbar is created.	
5	To add commands from another toolbar to your new toolbar, see step 8. To add menu commands to your toolbar, click on the Commands tab. This forces the new toolbar behind the Customize dialog, so you might have to drag the Customize dialog to the side to find your new toolbar.	
6	Find the command to add to your toolbar in the Commands list. The Categories list on the left represents the Enterprise Architect menu structure; the Commands list updates each time you click on a different category.	
7	Drag the selected command from the list into the new toolbar. If you selected the Show text labels checkbox, your toolbar should now look	

Step	Action	See also
	<p>like this:</p>  <p>If you did not select the Show text labels checkbox, your toolbar should look like this:</p> 	
8	<p>If the command you require is on another toolbar, display that toolbar and drag it to position it near your new toolbar.</p> <p>Press (Ctrl) and drag the required icon for the command from the existing toolbar onto your new toolbar.</p>	

Notes

- You can select the **Show text labels** checkbox to display textual descriptions of toolbar items
- You can add as many commands to your toolbar as required; your new toolbar behaves the same way as other toolbars - you can position it next to the other toolbars at the top of the application workspace, dock it to the side of the workspace or close it
- On the Customize dialog, you can customize all the listed toolbars except for the Format Tool toolbar (which is actually tied to displayed diagrams)

Learn more

- [Customize Commands](#)^[154]

2.5.6.1.3 Custom Tools

The Tools tab on the Customize dialog provides a means of extending the power of the Enterprise Architect desktop.

From this toolbar you can configure custom tools and make them accessible from the **Main Menu**. You can create menu options that hyperlink to different applications, compilers, batch scripts, automation scripts, URLs or documentation.

How to

To add and configure custom tools, follow the steps below

Step	Action	See also
1	Select the Tools Customize menu option. The Customize dialog displays.	
2	Click on the Tools tab.	
3	Click on the New icon (to the left of the red X). A blank field displays in the Menu contents list.	
4	Type in the name of the tool as it should appear in the menu.	
5	In the Command field, type the name of the tool .exe file to use; the tool must be a valid filename.	
6	Add any arguments required by the tool, and specify an initial directory if required.	Opening External Tools ^[157] Passing Parameters to External Applications ^[159]
7	Close the Customize dialog. Your tool should have now been added to the Tools menu.	

2.5.6.1.3.1 Open External Tools

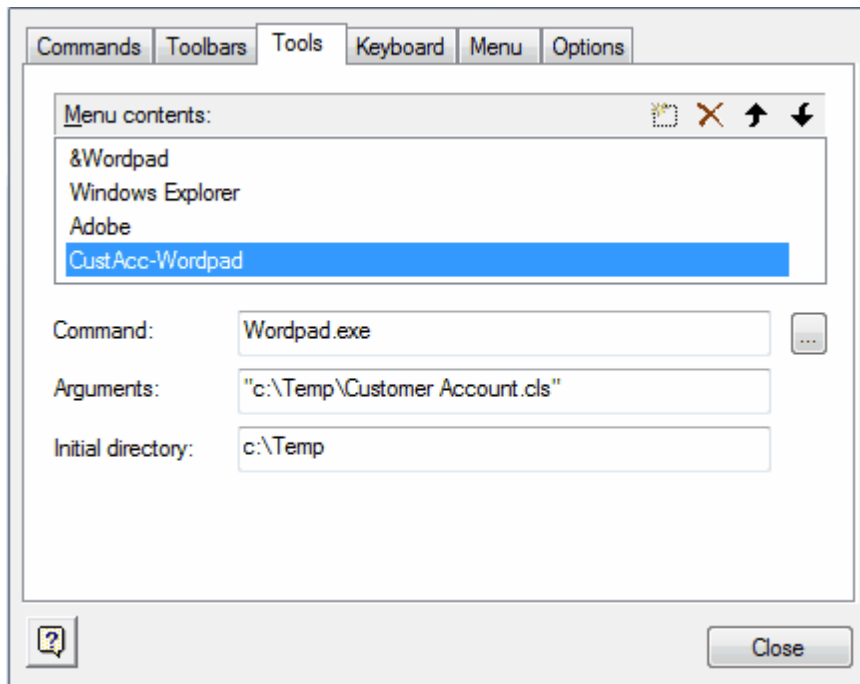
When configuring custom tools in Enterprise Architect, you can specify a file to be opened by the external application.

Select the **Tools | Customize** menu option. The Customize dialog displays; click on the Tools tab. Now you can:

- Specify a [custom tool](#) ^[156](application) using the **Command** field
- Define a file to open or [parameters to pass](#) ^[159]to this application, using the **Arguments** field

Example 1

This example opens the file `c:\Temp\Customer Account .cls` using Wordpad. If you save from within Wordpad the initial directory is `c:\Temp`.

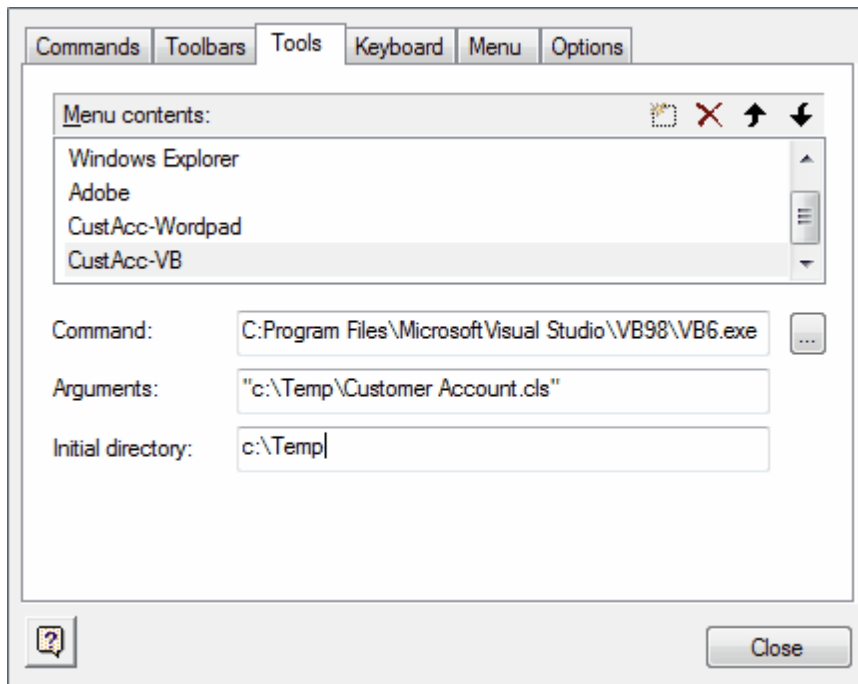


If there are any spaces in the paths in the **Command**, **Arguments** or **Initial directory** fields, you must enclose the whole path in double quotes.

" c : \ Temp \ Cust omer Account . cls " must have quotes but c : \ Temp \ Cust omer Account . cls does not have to have quotes.

Example 2

This example opens the file c : \ Temp \ Cust omer Account . cls using VB. As VB is not installed with the operating system, the whole file path for VB must be included in the **Command** field; you can select this using the (...) (Browse) button to locate the VB executable. If you save from within VB the initial directory is c : \ Temp .



2.5.6.1.3.2 Pass Parameters to Applications

When configuring custom tools in Enterprise Architect, you can pass parameters to the application.

Select the **Tools | Customize** menu option. The Customize dialog displays; click on the Tools tab. Now you can:

- Specify a custom tool (application) using the **Command** field
- Define a file to open or parameters to pass to this application using the **Arguments** field

The available parameters for passing information to external applications are:

Parameter	Description	Notes
\$f	Project Name	For example, C: \ p r o j e c t s \ E A e x a m p l e . e a p
\$F	Calling Application (Enterprise Architect)	Enterprise Architect
\$p	Current Package ID	For example, 144
\$P	Package GUID	GUID for accessing this package
\$d	Diagram ID	ID for accessing associated diagram
\$D	Diagram GUID	GUID for accessing associated diagram

Parameter	Description	Notes
\$e	Comma separated list of element IDs	All elements selected in the current diagram
\$E	Comma separated list of element GUIDs	All elements selected in the current diagram

Learn more

- [Custom Tools](#)^[156]
- [Opening External Tools](#)^[157]

For more information on using the Automation Interface, visit www.sparxsystems.com/AutIntVB.htm.

2.5.6.1.4 Customize Keyboard

The Keyboard tab on the Customize dialog enables you to configure shortcuts used to access main menu options. This is convenient for creating additional shortcut keys or for changing the current configuration to match your work habits or other applications.

Access **Tools | Customize**

How to

To modify a keyboard shortcut

Step	Action	See also
1	Select the Customize menu option. The Customize dialog displays.	
2	Click on the Keyboard tab. In the Category field click on the drop-down arrow and select the menu containing the command to modify.	
3	In the Command field, click on the drop-down arrow and select the command. The current shortcut key (if any) for the command is displayed in the Current Keys field.	
4	Move the cursor to the Press New Shortcut Key field and press the required shortcut key(s) for this command. Press the actual keys to use; for example, to use (F5) press the (F5) key, don't type F then 5 . The Assign button might be disabled if the selected keyboard shortcut is already used for another command; if this occurs the already existing command is	

Step	Action	See also
	shown and you must select a different shortcut key.	
5	Once you have selected an available shortcut, click on the Assign button to apply the change.	
6	Adding a keyboard shortcut to a command that already has a keyboard shortcut creates multiple shortcuts for a single command. You can remove the previous shortcut by selecting it and clicking on the Remove button.	

Notes

- Modified shortcut keys are stored in the registry, so they only affect the current user
- You can revert to the default shortcut keys at any time, by clicking on the **Reset All** button

2.5.6.1.5 Customize Menu

The Menu tab on the Customize dialog enables you to customize the appearance of your menus.

Option	Description	See also
Application Frame Menus	Currently the Show Menus For feature is disabled as Enterprise Architect is not an MDI application.	
Context Menus	Currently this feature is disabled.	
Menu Animations	The following menu animations can be selected from the Menu animations drop-down list: <ul style="list-style-type: none"> • None • Unfold • Slide • Fade • [Default] 	
Menu Shadows	Menu shadows can be toggled on or off by selecting or clearing the Menu shadows checkbox.	
Remove Menu Options	Some menu options might not be of relevance to you. If you prefer not to display such options, follow the steps below 1. Whilst the Menu tab of the Customize dialog is displayed, click on the appropriate chain of options in the main menu bar to display the	

Option	Description	See also
	<p>option to delete.</p> <p>2. Right-click on the option and select the Delete option from the context menu.</p>	

Notes

- When you perform a major upgrade of Enterprise Architect (such as from release 8.0 to 9.2) the menus are reset and deleted options are replaced.

2.5.6.1.6 *Customize Options*

The Options tab on the Customize dialog enables you to customize the appearance of toolbar items.

You can toggle the following options by selecting or clearing the checkboxes:

- **Show Screen Tips on toolbars**
- **Show shortcut keys in Screen Tips**
- **Use Large Icons**

2.5.6.2 *Visual Styles*

You can configure the overall look and feel of Enterprise Architect to suit your working environment. Options include various Microsoft Office and Visual Studio styles, or themes.

Access **View | Visual Style | Select Visual Style**

How to

To reset the appearance of Enterprise Architect

Step	Action	See also
1	Select the Select Visual Style menu option. The Application Look dialog displays.	
2	Select the required style from the list. If you select the Microsoft Office 2007 or Microsoft Office 2010 radio buttons, you can also select from a number of base-color options.	
3	To try out styles, click on the Apply button. To set the style and resume work, click on the OK button.	

Notes

- You can also enable customization of toolbars and menus, and animate auto-hidden windows

Learn more

- [Customize Menu](#)^[167]
- [Autohide Windows](#)^[137]

2.5.6.3 Manage Workspace Layout

The **Manage Workspace Layouts** icon displays the Workspace Layout dialog, which lists the currently-available user-defined and system layouts.

In the Existing Workspace Layout panel at the bottom of the dialog, the **Based On:** field identifies the defined layout that the current workspace layout was derived from - you might have moved or closed windows since applying that layout. The highlighted (*Copy of existing Workspace Layout*) at the top of the Workspace Layout dialog is a capture of the workspace layout immediately before you opened the dialog.

Action	Description	See also
Change Layout	<p>You can now change the layout in the Existing Workspace Layout panel to:</p> <ul style="list-style-type: none"> • The original layout (as identified by the Based On: field), discarding any changes you might have made • The <i>Copy of existing Workspace Layout</i>, preparatory to saving the changes in a new named layout, or • One of the other named layouts <p>To change the layout in use, either:</p> <ul style="list-style-type: none"> • Double-click on the required layout name • Click on the layout name and click on the Apply button or OK button, or • Right-click on the layout name and select the Apply context menu option <p>When the layout in use changes, the layout name in the toolbar Workplace Layout Selection field also changes.</p>	
Copy Layout	<p>To copy a layout, either:</p> <ul style="list-style-type: none"> • Change the layout in use to the required layout and click on the Save As button, or • Right-click on the layout name and select the Save As context menu option <p>The Save Custom Workspace Layout dialog displays.</p>	

Action	Description	See also
	<p>In the Custom Workspace Layout Name field, type a name for the layout. Again, by selecting an existing name you can change an existing layout to something different. Click on the Save button.</p> <p>If you already have tailored windows or views that you want to include in your selected layout, select the Include active custom views checkbox.</p>	
Delete Layout	To delete a workspace layout, right-click on the layout name and select the Delete context menu option. Enterprise Architect prompts you to confirm or cancel the deletion.	

Workspace Layout Selection:

Enterprise Architect provides a number of layouts of windows and toolbars to suit particular areas of work, such as Requirements Management, Code Engineering and Debugging. As described above, you can also add your own layouts to the drop-down list; these custom layout names have a preceding asterisk in the drop-down list for the field (the system-provided layouts are not marked with an asterisk).

To switch to another layout, click on the drop-down arrow and click on the required layout.

2.5.7 Other Windows

Most of the windows in Enterprise Architect have a specific, task oriented purpose. Two windows have broader functions.

Window	Detail	See also
System Output Window	This option displays the data that Enterprise Architect generates during each of a range of processes.	The System Output Window ^[164]
Web Browser	This option enables you to search for and use internet facilities within your Enterprise Architect work area.	The Web Browser ^[165]

2.5.7.1 The Output Window

The System Output window is used to display information generated by internal Enterprise Architect processes, or by Add-Ins and other third-party tools. This window can provide useful information during long-running processes and batch operations. Many validation processes within Enterprise Architect write out their results to this window.

Access

- **View | System Output** (Ctrl+Shift+6)
- Element Context Menu
- Notes Window Context Menu

Use to

- Validate information
- Progress information during launch of external processes
- Command line output from Build and Test
- Parse errors generated during import of various types of files
- Auditing Information *
- Re-docking the [Model Search](#)^[611] results into the Output window

How to

Right-click on an item and select context menu options to:

- Copy the selected item to the clipboard
- Copy all items to the clipboard
- Save the output to an external file
- Clear the output from the window
- Rollback selected [Get All Latest](#)^[357] updates to a version controlled package

Drag the selected items/text out of the System Output window and onto a diagram to quickly create a named element.

Double-click on model validation errors or parsing errors to display the source of the error.

Switch between tabs to view different categories of information from different tools.

Notes

- The System Output window can also be used by [Add-Ins](#)^[2583], if they are configured to do so.
- * (Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect) The [Audit History](#)^[392] tab of the Output window shows a history of changes to any element or connector selected from the Audit View, the Package Browser, the Diagram List, the Project Browser or the current diagram ([Auditing](#)^[382] must be turned on)

2.5.7.2 The Web Browser

The Web Browser displays as a tab of the central work area, like the Start Page, Model Search, Package Browser and Diagram View. It provides access within Enterprise Architect to internet facilities such as email, websites and search engines.

To access the Web Browser:

- Press (**Ctrl+Alt+W**), or
- Select the **View | More Project Tools | Internal Web Browser** menu option

The Web Browser opens at the default home web site; you define the default home website, search engine and email exchange address on the General page of the Options dialog.

To access the:

- Email exchange server, click on the 'envelope' icon in the toolbar; the email login window displays
- Web search engine (such as *Google*), click on the 'spyglass' icon in the toolbar; the search engine screen displays
- Home web site, after displaying other web pages, click on the 'house' icon in the toolbar

To go directly to another website or email server (your internet security permitting), in the **Address** field type or select the website http address and click on the **Go** button.

2.5.8 Keyboard Shortcuts

The table below lists the default keyboard shortcut functions within Enterprise Architect.

Notes

- You can also display the key combinations on the Help Keyboard dialog (**Help | Keyboard Accelerator Map**; click on the drop-down arrow in the **Category** field and select the appropriate category)
- There are some additional shortcuts using the keyboard and mouse in combination; see the [Keyboard-Mouse Shortcuts](#)^[174] topic
- If necessary, you can change these keyboard shortcuts using the Keyboard tab of the [Customize](#)^[160] dialog

Action	Shortcut	Category
Create a new Enterprise Architect project	Ctrl + N	File
Open an Enterprise Architect project	Ctrl + O	File
Open Source File	Ctrl + Alt + O	File
Reload the current project ^[242]	Ctrl + Shift + F11	File
Print the active diagram	Ctrl + P	File
Undo Change	Ctrl + Z	Edit
Redo Change	Ctrl + Y	Edit

Action	Shortcut	Category
Add a single element to the clipboard list	Ctrl + Space	Edit
Paste element as metafile from clipboard	Ctrl + Shift + Insert	Edit
Paste element as new	Ctrl + Shift + V	Edit
Paste element(s) from the clipboard	Shift + Insert	Edit
Bookmark current element with red marker	Shift + Space	Edit
Delete selected element(s) in diagram	Delete or Ctrl + D	Edit
Delete selected element(s) from model (through diagram OR Project Browser)	Ctrl + Delete	Edit
Search for items in the project	Ctrl + F	Edit
Set focus to current window	Ctrl + Shift + 0	Window
Autohide the current window	Ctrl + Shift + F4	Window
Hide the current window	Ctrl + F4	Window
View Project Browser	Alt + 0	View
View Properties window	Alt + 1	View
View Project Information window	Alt + 2	View
View Testing window	Alt + 3	View
View Maintenance window	Alt + 4	View
Display Toolbox	Alt + 5	View

Action	Shortcut	Category
View Resources window	Alt + 6	View
View Source Code window	Alt + 7	View
View Debug Workbench	Alt + 8	View
View Notes window	Ctrl + Shift + 1	View
View Element Relationships window	Ctrl + Shift + 2	View
View Rules and Scenarios (Requirements and Constraints) window	Ctrl + Shift + 3	View
View Traceability window	Ctrl + Shift + 4	View
View Tagged Values window	Ctrl + Shift + 6	View
View Project Management window	Ctrl + Shift + 7	View
View Output window	Ctrl + Shift + 8	View
View Learning Center	Alt + F1	View
View Pan & Zoom Window	Ctrl + Shift + N	View
View Model Search	Ctrl + Alt + A	View
View Package Browser (or Diagram List)	Ctrl + Alt + R	View
Open Team Review	Ctrl + Alt + U	View
Display Web Browser	Ctrl + Alt + W	View

Action	Shortcut	Category
View Element Browser	Alt + 9	View
Add new package to project	Ctrl + W	Project
Add new diagram to package	Ctrl + Insert	Project
Add new element to package	Ctrl + M	Project
Create RTF documentation	F8	Project
Generate HTML Report	Shift + F8	Project
Generate Diagrams-only Report	Ctrl + Shift + F8	Project
Generate package source code	Ctrl + Alt + K	Project
Synchronize package contents	Ctrl + Alt + M	Project
Import source directory	Ctrl + Shift + U	Project
Manage Analyzer Scripts on Execution Analyzer window	Shift + F12	Project
Build	Ctrl + Shift + F12	Project
Test	Ctrl + Alt + T	Project
Run	Ctrl + Alt + N	Project
Deploy	Ctrl + Shift + Alt + F12	Project
Debug Run	F6	Project
Step Into	Shift + F6	Project

Action	Shortcut	Category
Step Over	Alt + F6	Project
Step Out	Ctrl + F6	Project
Debug Stop	Ctrl + Alt + F6	Project
Transform selected elements	Ctrl + H or Ctrl + Alt + F	Project
Transform current package	Ctrl + Shift + H	Project
Validate Selected	Ctrl + Alt + V	Project
Manage locks applied by current user	Ctrl + Shift + L	Project
Configure package control	Ctrl + Alt + P	Project
Import package from XMI	Ctrl + Alt + I	Project
Export package to XMI	Ctrl + Alt + E	Project
Import and export to CSV files	Ctrl + Alt + C	Project
Manage Baselines	Ctrl + Alt + B	Project
Diagram properties	F5	Diagram
Save	Ctrl + S	Diagram
Save image to file	Ctrl + T	Diagram
Save image to clipboard	Ctrl + B	Diagram
Visible Relations	Ctrl + Shift + I	Diagram

Action	Shortcut	Category
Locate in Project Browser	Shift + Alt + G	Diagram
Repeat last element	Shift + F3 or Ctrl + click	Diagram
Repeat last connector	F3	Diagram
Element Properties	Alt + Enter	Element
Add Tagged Value	Ctrl + Shift + T	Element
Linked Document	Ctrl + Alt + D	Element
Display Attribute Properties dialog	F9	Element
Display Operation Properties dialog	F10	Element
Space elements evenly horizontally	Alt + -	Element
Space elements evenly vertically	Alt + =	Element
Add attribute	Ctrl + Shift + F9	Element
Add operation	Ctrl + Shift + F10	Element
Add other type	Ctrl + F11	Element
Auto-size selected elements	Alt + Z	Element
Generate code from element	Ctrl + G or F11	Element
Move element by increments	Shift + ↑ , ↓ , → , ←	Element
Resize selected element	Ctrl + ↑ , ↓ , ← , →	Element

Action	Shortcut	Category
Align bottom edges of selected elements	Ctrl + Alt + Down	Element
Align top edges of selected elements	Ctrl + Alt + Up	Element
Align selected elements on left boundaries	Ctrl + Alt + Left	Element
Align selected elements on right boundaries	Ctrl + Alt + Right	Element
Configure element default appearance	Ctrl + Shift + E or F4	Element
Edit selected	F2	Element
Manage embedded elements	Ctrl + Shift + B	Element
Insert new feature after current selection	Insert	Element
Locate in browser	Alt + G	Element
New element	Ctrl + M	Element
View source code in default editor	Ctrl + E or F12	Element
Operation	F10	Element
Override inherited features	Ctrl + Shift + O	Element
Configure element properties	Alt + Enter	Element
Select alternative image	Ctrl + Shift + W	Element
Specify which element features are visible on a diagram	Ctrl + Shift + Y	Element
Set element parent or implement interface(s)	Ctrl + I	Element

Action	Shortcut	Category
Set references to other elements and diagrams	Ctrl + J	Element
Create Workbench Instance	Ctrl + Shift + J	Element
Locate diagrams where element is used	Ctrl + U	Element
View Properties dialog	Enter	Element
Check project data integrity	Shift + F9	Tools
Configure system options	Ctrl + F9	Tools
Spell check current package	Ctrl + Shift + F7	Tools
Spell check model	Ctrl + F7	Tools
Spell check Notes text	F7	Object notes
Edit code generation templates	Ctrl + Shift + P	Settings
Edit transformation templates	Ctrl + Alt + H	Settings
Make text bullet list item	Ctrl + . (full stop)	Object notes
Make text numbered list item	Ctrl + 1	Object notes
Make text bold	Ctrl + B	Object notes
Make text italic	Ctrl + I	Object notes
Make text underlined	Ctrl + U	Object notes
Insert date and time in Notes text	F5	Object notes

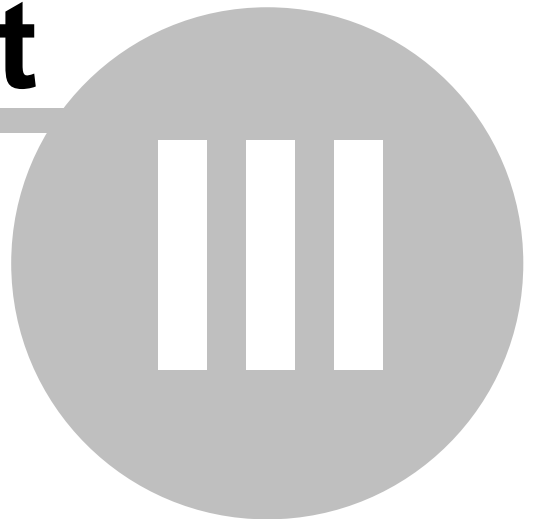
Action	Shortcut	Category
Select line of text in Notes text	F8	Object notes
Copy text	Ctrl + C	Everywhere
Paste text	Ctrl + V	Everywhere
Cut text, or element in diagram	Ctrl + X	Everywhere

2.5.8.1 Keyboard-Mouse Shortcuts

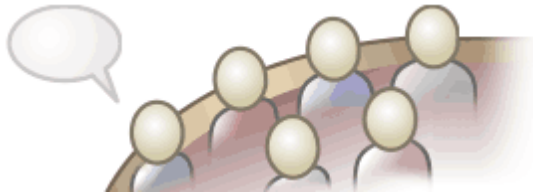
The following table lists certain operations you can perform quickly using a combination of keyboard keys and the computer mouse.

Action	Shortcut
Create an element of the same type as was last created on the diagram.	Ctrl + click
Display the element Properties dialog for the element shown in the scenario Context References tab.	Ctrl + click
Select a number of individual objects for an operation.	Ctrl + click each object
Select a range of objects for an operation.	Shift + click the first and last object in the range
Move the selected elements along the horizontal axis or vertical axis.	Shift + hold left mouse button and drag
Create a bend in and change the direction of the connector line.	Shift + hold left mouse button and drag
Move the selected elements in any direction, including diagonally.	Alt + hold left mouse button and drag
Zoom in to or zoom out of the diagram.	Ctrl + mouse scroll wheel
Pan the diagram horizontally or vertically.	Ctrl + Shift + hold left mouse button and drag

Part






3 Projects and Teams



Enterprise Architect helps you to create projects for development under a [range of work conditions](#)^[178], from single user/local access through to multiple-role teams working in a distributed environment. You both protect and manage the model data itself, and communicate information on the data in the form of documentation and reports, using facilities such as those listed below.

Topics

Topic	Link
Creating a project in a simple, file-based repository (.EAP file)	File-Based Repositories ^[189]
Creating the project in one of a range of DBMS repositories - these provide for larger models with more concurrently connected users (Corporate and extended editions)	Server-Based Repositories ^[193]
Tools for enabling team or multiple-user development in the project	Team Development ^[240]
Tools for managing change within the model	Change Management ^[311]
A variety of tools for managing modeling and project activities	Project Management ^[447]
Maintaining the integrity of the project data	Project Maintenance ^[534]
Sharing the reference data used across the project, between models and between projects	Sharing Reference Data ^[305] Reference Data ^[981]
Enabling each user to configure their personal preferences for how project tools display and behave on their workstation	Local Options ^[542]
You can have recorded discussion and communication of decisions using the Team Review	Team Review Tools ^[279]
Managing work and resources allocated to tasks	Project Task Allocation ^[472]

Topic	Link
Communicating and collaborating on model development through the Model Mail facilities	Personal Information  <small>[486]</small>
Tracking important project events, resource issues and allocations through the Project Calendar	Project Calendar  <small>[502]</small>
Documenting your model through RTF or HTML reports, which you can tailor to your purposes	Reporting  <small>[2249]</small>

3.1 Introduction



An Enterprise Architect project is stored in a data repository.

Topics

Topic	Detail	See also
Project Files	<p>.EAP Files</p> <p>In Enterprise Architect Desktop and Professional editions, you store a project in a single file with a .EAP extension. A .EAP file is a Microsoft JET database, so you can also open it using MS Access 97, 2000 or 2003, or any other reporting tool that can work with JET databases.</p> <p>DBMS Repositories</p> <p>In Enterprise Architect Corporate Business and Software Engineering, Systems Engineering and Ultimate editions, you can use a suitable DBMS database for project files.</p> <p>DBMS project files have the same logical structure as .EAP files, but must be connected to using ADO/ODBC (see <i>Connect to a Data Repository</i>, below).</p>	<p>File Based Repositories ^[189]</p> <p>Server Based Repositories ^[193]</p>
Creating Project Files	<p>On creating a new project, the Model Wizard enables you to create a model containing various Model Packages.</p> <p>You can also add Model Packages to a project from the Project Browser by:</p> <ul style="list-style-type: none"> • Right-clicking on an existing model and selecting the New Model or Add a New Model using Wizard context menu options • Right-clicking on a package and selecting the Add Add a New Model using Wizard context menu option • Clicking on an existing model, pressing (Insert) and selecting the New Model or Add a New Model using Wizard context menu options • Clicking on a package, pressing (Insert) and 	<p>File Based Repositories ^[189]</p> <p>Model Wizard ^[660]</p>

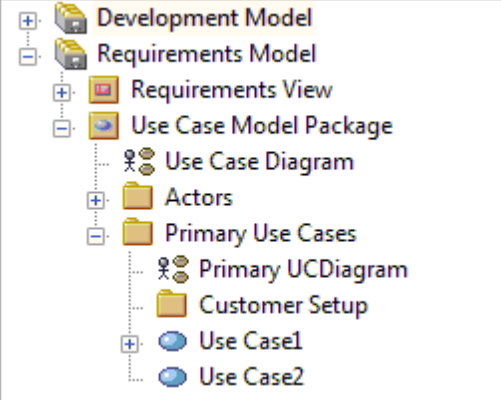
Topic	Detail	See also
	selecting the Add a New Model using Wizard context menu option	
Opening Existing Projects	There are various ways to open a project in Enterprise Architect; new users are advised to explore the <i>EASample</i> file supplied with Enterprise Architect.	Open a project ^[187]
Connect to a Data Repository	Enterprise Architect enables you to connect to any of the following data repositories: <ul style="list-style-type: none"> • MS Access 97, 2000 and 2003 (in all editions - .EAP files are stored in Microsoft JET databases) • Access 2007 • SQL Server 2000, 2005 and 2008 • MySQL • Oracle 9i, 10g or 11g • PostgreSQL • MSDE • Adaptive Server Anywhere • Progress OpenEdge <p>To create a new data repository, you must first create a new database with the DBMS management software, then run supplied scripts to create the logical structure.</p> <p>You should then use Enterprise Architect data transfer functions to move a project from a .EAP or DBMS model into the new project.</p>	Connect to a Data Repository ^[209]

3.1.1 Projects Defined

An Enterprise Architect Project is a repository for storing, manipulating and managing one or more Models. A single repository can contain many models, and a repository can be either file based or hosted in a larger DBMS system.

Guide:

Topic	Detail	See also
Project	A project can contain a single model, or a number of models, each of which defines a particular system or process. A model contains the diagrams, elements, relationships and associated metadata that define the structure and function of the system or process.	Open a Project ^[187] Model Shortcuts ^[183] Project Management ^[447] Team Development ^[240]

Topic	Detail	See also
	<p>These components are organized into a hierarchy of packages, which help to group and manage related components.</p>	
<p>Model Package</p>	<p>Different aspects of the process or system - or their development - are defined by Model Packages, which you generate from templates specifically structured to support the aspects that the Model Packages represent, such as requirements or deployment.</p> <p>You can generate these templated packages at any level of the hierarchy, but as they are created with their own content they are more useful at the top levels.</p>	<p>Model Wizard^[660] Model Templates^[662]</p>
<p>View</p>	<p>The top-level packages in a model can also be Views, which represent partitions of the model that you define yourself.</p> <p>You can start with standard Views such as Class or Component, or create whatever partitions are appropriate to your model.</p>	<p>Views^[676]</p>
<p>Example Project Structure</p>	 <p>The screenshot shows a tree view of a project structure. At the top are 'Development Model' and 'Requirements Model'. Under 'Requirements Model' are 'Requirements View' and 'Use Case Model Package'. Under 'Use Case Model Package' are 'Use Case Diagram', 'Actors', and 'Primary Use Cases'. Under 'Primary Use Cases' are 'Primary UCDiagram', 'Customer Setup', 'Use Case1', and 'Use Case2'.</p> <p>Each View or Model Package contains packages; <i>Use Case Model Package</i> contains:</p> <ul style="list-style-type: none"> • Actors and • Primary Use Cases <p>It also contains the diagram Use Case Diagram, which could be an overview of the package structure or function.</p> <p>Each package itself can contain one or more diagrams, one or more packages, and several elements; the <i>Primary Use Cases</i> package contains the:</p> <ul style="list-style-type: none"> • Primary UCDiagram • Customer Setup package • Use Case1 element 	

Topic	Detail	See also
	<ul style="list-style-type: none"> Use Case2 element <p>Each subordinate package also contains diagrams, elements and (if necessary) further packages; the elements are related by connectors created in the diagrams, and each element and connector has properties, attributes, operations and extensions defined in the respective Properties dialogs.</p>	
Storage	A project can be a .EAP file in an MS Access database or (in the Enterprise Architect Corporate, Business and Software Engineering, System Engineering and Ultimate editions) a structure of files in a database management system such as MySQL or Oracle.	File Based Repositories ^[189] Server Based Repositories ^[193]

3.1.2 Open a Project

An Enterprise Architect project is used for storing and managing the components of one or more UML models. The Desktop and Professional versions of Enterprise Architect use an MS JET database as the model repository. If you are using the Corporate edition (or above), you can also use DBMSs such as Oracle and MySQL to host the model repository.

Access [File | Open Project](#)

Use to

- Create a new Enterprise Architect project file (the model repository is created as an MS Jet database file)
- Browse for an existing project file to open
- Specify connection details for opening a project hosted on a DBMS
- Select a recent project to open
- Remove items from the list of recent projects

Reference

Field	Usage	See also
New Project...	Create a new project, as a .EAP file.	File Based Repositories ^[189]
Browse for Project	Open a file browser dialog, to select an existing EAP file; the EAP file can be a project file or a shortcut to a project hosted on a DBMS.	Model Shortcuts ^[183]
Connect to Server	Specify connection details for opening an Enterprise Architect project that is hosted on a DBMS.	Server-Based Repositories ^[193]

	<p>If you double-click on the option, the Windows Data Link Properties dialog displays; start to define the connection to the project's server-based repository.</p> <p>If you click on the drop-down arrow and select the Connection Wizard option, the Data Link Properties dialog again displays.</p> <p>If you click on the drop-down arrow and select the Connection String option, the Connection String dialog displays; type in or paste the connection string and click on the OK button to connect directly to the project.</p>	
Recent Projects	<p>Show a list of the ten most recently opened projects.</p> <p>This list is also displayed:</p> <ul style="list-style-type: none"> • On the Start Page, under the Recent heading • As a drop-down menu under the Open Project button on the Default Tools toolbar • As part of the File menu <p>On the Open Enterprise Architect Project dialog, each project name is shown with its access path.</p> <p>You can edit the access path by right-clicking on it and selecting the Edit Connection String context menu option; make the changes on the Connection String dialog.</p>	
Remove Selection from List	Remove the currently selected project from the Recent Projects list.	
Show this Dialog at Startup	Select whether or not to show the Open Enterprise Architect Project dialog upon starting Enterprise Architect.	
Open	Open the currently selected project in the Recent Project list.	
Cancel	Close this dialog without any further action.	
Help	Display this Help topic.	

Notes

- Use of a DBMS for hosting the model repository is only available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect
- Enterprise Architect supports the following DBMS products for hosting model repositories; [SQL Server](#)

[\[195\]](#), [MySQL](#) [\[195\]](#), [Oracle 9i, 10g or 11g](#) [\[197\]](#), [Postgre SQL](#) [\[218\]](#), [ASA](#) [\[219\]](#), [MSDE Server](#) [\[221\]](#) and [Progress OpenEdge](#) [\[221\]](#)

Learn more

- [Projects Defined](#) [\[179\]](#)
- [File Based Repositories](#) [\[189\]](#)
- [DBMS Repositories](#) [\[193\]](#)
- [Team Development](#) [\[240\]](#)

3.1.3 Project Shortcuts

Enterprise Architect enables you to create a desktop shortcut (or *proxy* file) to an Enterprise Architect project (DBMS or file-based). Each shortcut is a file containing the connection string for the model.

However, the shortcut also defines *views* that Enterprise Architect should open when it opens the model, as outlined below.

Topics

View	Detail	See also
Diagrams	Define one or more specific diagram(s) to be loaded on opening the project.	
Model Search	Open the Model Search with a specific text string and search type. For searches operating on the current tree selection, a diagram in the target package must be opened first. If you use a custom SQL search, the SQL must include <i>ea_guid AS CLASSGUID</i> and the object type. You specify a single Model Search to open.	Model Search [611]
The Relationship Matrix with a saved profile	Open the Relationship Matrix with a saved profile. You specify a single Relationship Matrix profile to open.	Relationship Matrix [639]
The default Team Review	Open the default Team Review document. You specify the Team Review once.	Team Review Tools [279]
Working Set	Open a working set. You specify a single Working Set to open. This is very similar to the shortcut itself, opening a defined set of diagrams and views. However, the working set can also open source code editors, therefore widening the capabilities of the shortcut alone. Working sets make it easy to customize the main views you	Working Sets [498]

View	Detail	See also
	<p>want to open without having to resave your project short cuts. The working set is easy to tailor to your changing requirements, adding greater flexibility to any short cut that calls the working set</p> <p>Also, working sets provide the capability of storing the currently-engaged files when closing a model, and reopening them in the context in which you were last using them. The basic project shortcut can also do this, but once the work environment is captured the shortcut returns to the same set up each time you use it. The working set always captures the current work environment each time the model closes.</p> <p>This option can be very useful in, for example, a coding environment when you want to return to the last files you were editing.</p>	
Workspace Layout	<p>Apply a selected Workspace Layout, which opens and sets out the appropriate screens and windows for a specific area of work, such as model simulation.</p> <p>You specify a single Workspace Layout to open. However, by creating and using separate shortcuts you can open Enterprise Architect in exactly the configuration you need for the work you want to do - modeling, coding, or debugging, for example.</p>	Workspace Layouts ^[149]
An example shortcut	<p>You might create a shortcut to open, in sequence:</p> <ul style="list-style-type: none"> • A Development module • The Model Search for a simple search on the term <i>Issue</i> • The module <i>Issues</i> diagram • The module <i>Changes</i> diagram <p>Enterprise Architect opens the appropriate windows in the sequence in which you list the options, displaying the last view in the list as the active view.</p> <p>In this example, the project opens with the Enterprise Architect work area showing the two diagram tabs and the Model Search tab, and with the <i>Changes</i> diagram displayed in the Diagram View.</p>	

Notes

- If specified, the shortcut views override any default diagram defined for the model or current user
- A shortcut does not affect the original Enterprise Architect .exe file or icon, or any other shortcut you might have defined; you can use all of these independently
- If you are using a database repository other than MS Access 97, 2000 or 2003, you can configure the

shortcut to **encrypt the password** used to set up the connection between Enterprise Architect and the repository; the Enterprise Architect user does not have the real password, thereby preventing them from accessing the repository using other tools such as Query Analyzer or SQLPlus

Learn more

- [Create Project Shortcut](#)^[185]
- [Capture Current Work Environment](#)^[187]
- [Encrypt Repository Password](#)^[188]
- [Default Diagram](#)^[105]
- [Creating Search Filters](#)^[623]

3.1.3.1 Create Project Shortcut

You can create a shortcut to an Enterprise Architect project (either a DBMS project or file based project). The shortcut can specify additional windows and diagrams to open up automatically every time the shortcut is run.

Shortcuts are stored with a .EAP extension, but are actually small text files that tell Enterprise Architect what project to open and what initial views and windows to display.

Access: **File | Save Project Shortcut**

Use to

- Specify a working environment in advance, for other users

How to

To create a project shortcut

Step	Action	See also
1	Open Enterprise Architect.	
2	Open the required project.	
3	Select the Save Project Shortcut menu option. The Save Project Shortcut dialog displays.	
4	Click on the (...) (Browse) button at the end of the Target File field. The Save Project As dialog displays.	
5	Browse for the appropriate file location and, in the File name field, type an appropriate filename.	

Step	Action	See also
	All shortcuts are .EAP files, regardless of whether the model itself is a .EAP file or a DBMS model.	
6	Click on the Save button to return to the Save Project Shortcut dialog.	
7	<p>Click on the Add Other button and select the required option to define:</p> <ul style="list-style-type: none"> • A diagram to open • A Relationship Matrix profile to open • The Team Review • A Model Search to perform • A working set to apply • A workspace layout to apply 	
8	<p>The appropriate browser or dialog displays to define the view to display; enter the details and click on the OK button.</p> <p>The view is added to the Actions when model is opened field; the entry is automatically selected, with a tick in the checkbox.</p>	
9	Repeat steps 7 and 8 for as many additional views as you require.	
10	<p>Review the items in the Actions when model is opened field.</p> <ul style="list-style-type: none"> • If you decide not to have an item in the shortcut, deselect its checkbox • If you want to clear all selected items, click on the Include None button <p>Unselected entries are deleted when you save the shortcut.</p>	
11	<p>If you decide to change the sequence and/or make a different view display first in the Diagram View:</p> <ul style="list-style-type: none"> • Click on the appropriate entry • Click on the 'Up Hand' or 'Down Hand' buttons 	
12	Click on the OK button to save the shortcut.	

Notes

- When you subsequently open the Save Project Shortcut dialog, it lists the currently-opened views in the order in which they were opened; you can add further views or remove them from the shortcut

Learn more

- [Capture Current Work Environment](#)^[187]
- [Encrypt Repository Password](#)^[188]

3.1.3.2 Capture Current Work Environment

You can capture the current Enterprise Architect work environment in your shortcut. This maintains the work environment from that point, and is useful if you intend to close and re-open the model and return to **the same** configuration many times.

If you simply want to return to **whatever** work environment you happen to be in each time you close the model, you should use a **work set** that captures the current environment, either by itself or as part of a broader project shortcut.

Access: **File | Save Project Shortcut**

Use to

- Capture the current Enterprise Architect work environment, to access the model at exactly the same point in exactly the same environment in your next work session

How to

To capture your current work environment

Step	Action	See also
1	Open Enterprise Architect.	
2	Open the required project and work in it.	
3	At the point at which you decide to capture your work environment in a shortcut, ensure that: <ul style="list-style-type: none"> • You have opened all diagrams you require • If necessary, you have opened the Team Review, Model Search (with appropriate search term and type) and/or Relationship Matrix (at the appropriate profile) • The view you want to resume work on is the last one opened 	
4	Select the Save Project Shortcut menu option. The Save Project Shortcut dialog displays, showing a list of actions derived from the views you currently have open.	
5	If you accessed Enterprise Architect via a shortcut, the Target File field displays the file location of that shortcut.	

Step	Action	See also
	Otherwise, click on the (...) (Browse) button at the end of the Target File field. The Save Project As dialog displays.	
6	Browse for the appropriate file location and, in the File name field, type an appropriate filename. All shortcuts are .EAP files, regardless of whether the model itself is a .EAP file or a DBMS model.	
7	Click on the Save button to return to the Save Project Shortcut dialog.	
8	In the Actions when model is opened field, click on the Include All button.	
9	If you also want to save current window positions click Add Other followed by Add Workspace Layout... The dialog shown allows you to select an existing layout or save the current layout.	Workspace Layouts [149]
10	Click on the OK button to save the shortcut.	

Learn more

- [Working Sets](#) [501]
- [Encrypt Repository Password](#) [188]

3.1.3.3 Encrypt Repository Password

If your model is developed on a DBMS repository, the Save Project Shortcut dialog has an **Encrypt Connection String** check box.

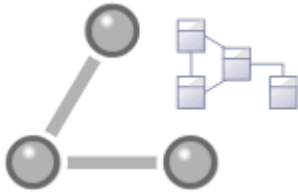
You can create the shortcut actions and, if necessary, select the checkbox to encrypt the database connection string.

You distribute the shortcut file to the database users who are to access the model. The users then have an encrypted string that prevents them from directly accessing the database using other tools.

Learn more

- [Create Project Shortcut](#) [185]

3.2 File Based Repositories



Having a light weight, portable, file-based repository for Enterprise Architect is ideal for those who want to quickly develop modeling suggestions and solutions and share the model information easily with others. All editions of Enterprise Architect support this type of repository in the form of .eap files in an Access database.

A default project file (*EABase.eap*) is supplied in the Enterprise Architect installer and is automatically stored in your Enterprise Architect installation directory. You can:

- Copy the EABase.eap file as the base for your own projects
- Create your own base projects as .eap files, and
- Copy existing operational projects

Topic	Detail	See also
Create .EAP Project Files	<p>Select one of:</p> <ul style="list-style-type: none"> • File New Project menu option • New Project option on the Open Enterprise Architect Project dialog, or • Create a New Project option on the Start Page <p>All of these options display the New Project dialog; select a directory and enter a file name for your project.</p> <p>Once the project has been saved, the Model Wizard displays, which makes a selection of Model Patterns available; select the Model Patterns to use.</p> <p>Enterprise Architect adds a model containing the selected Model Packages to the Project Browser.</p>	Model Wizard ^[660]
Default EABase.EAP Installation directories	<p>The default installation directories, depending on which version you have installed, are:</p> <ul style="list-style-type: none"> • Registered version: C:\Program Files\Sparx Systems\EA • Trial version: C:\Program Files\Sparx Systems\EA Trial • Lite version: C:\Program Files\Sparx Systems\EA Lite <p>Having copied the base project as a template for your own</p>	Copy a Base Project ^[190] Rename a Project ^[539]

Topic	Detail	See also
	project, you can rename it.	
Configure Project	Having created your project, you can set a range of project parameters to define defaults, tailor the project to particular coding languages, and ensure consistent development and use of the project.	The Settings Menu ^[127] Defaults and User Settings ^[153]
Create Custom Templates	You can customize any Enterprise Architect project as a template project with company standards, tutorials, frameworks and any other common piece of modeling already in-built; with careful planning you can save yourself many hours of work at project start-up.	Incorporate Model Templates ^[135]
Copy Existing Project	A Base project contains templates and reference data to enable you to develop your own project quickly. You can also copy an operational .EAP project file to enable separate development by team members, or to create an evaluation or distribution version of the project.	Copy Existing Projects ^[197]

Notes

- You can also add Model Packages to a project using the **New Model From Pattern** icon in the Project Browser toolbar

3.2.1 Copy a Base Project

You can copy an existing project as a new project. When using an existing project as the template for a new one, it is important to use this method, rather than simply copying the file using Windows Explorer. This process resets all the unique identifiers for packages and elements, so that your new project is truly unique - otherwise it is simply an exact copy of the source project.

Access **Start Page: Copy a Base Project**

Use to

- Create a project already largely set up, from a template project containing company standards, tutorials, frameworks and any other common piece of modeling
- Ensure all unique ID's (GUIDS) in a model are reset when starting a new model from an existing master

How to

Step	Action	See also
1	On the Enterprise Architect Start Page, click on the Copy a Base Project option. The Create New Enterprise Architect Project dialog displays.	Start Page ^[87]
2	In the Model Project field, select the project that is the base template. The field defaults to EABase.eap, the default base project provided in the Enterprise Architect installer, but you can click on the Browse button and locate any other custom project file to use.	Incorporate Model Templates ^[1355]
3	In the New Project field, click on the Browse button and select the file path for saving your project. If this is to be a shared project, store the file on a shared network resource such as a Network Server or Workgroup Server.	
4	To replace all GUIDs from the source model with fresh GUIDs in the new model, select the Reset New Project GUIDs checkbox. However, if the new project is based on one that is already under version control, we recommended that you deselect this checkbox to prevent duplication of packages when the <i>Get Latest</i> facility is used.	Package Version Control Menu ^[357]
5	Click on the Create Project button to create your project.	

3.2.2 Copy Existing Project

You can copy an operational .EAP project file to a new file location under a new name.

Access **File | Save Project Copy**

Use to

- Provide separate copies for individual team members
- Create an evaluation or distribution version of the project

How to

To create a copy of a project

Step	Action	See also
1	Select the Save Project Copy menu option. The Save Enterprise Architect Project Copy dialog displays.	
2	In the Target Project field, type the new project filename.	

Step	Action	See also
	You can either type in the file path as well, or click on the Browse button and browse for the required file path.	
3	If this copy is to be used for a different project, select the Reset New Project GUIDs checkbox. This ensures that all components of the new project have different IDs to their counterparts in the source project.	
4	Click on the Save As button to save the new project.	

3.3 Server Based Repositories



If you purchase the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions, you can upsize the .eap templates (such as EABase.eap) or any existing .eap project to a DBMS data repository.

A DBMS repository enables you to support larger models with more concurrently connected users.

Topic	Detail	See also
Supported DBMSs	<ul style="list-style-type: none"> • SQL Server 2000, 2005 or 2008 • MySQL 4 or 5 • PostgreSQL 7, 8 or 9 • Adaptive Server Anywhere 8 or 9, or SQL Anywhere 10, 11 or 12 • Access 2007 • Progress OpenEdge • MSDE or • Oracle 9i, 10g or 11g 	
Upsize Project to DBMS Repository	<p>To upsize the Enterprise Architect models (either existing or template) to use your selected DBMS:</p> <ol style="list-style-type: none"> 1. Install the DBMS software; how you do this is explained in your DBMS documentation and beyond the scope of the Enterprise Architect documentation. 2. Create a repository and run a script supplied by Sparx Systems to create the required tables. 3. Set up an ODBC driver to enable connection to the repository(not all DBMSs require this). 4. Open Enterprise Architect and perform a Project Integrity Check on the project .eap file (existing or template); this ensures the data is clean before uploading. 5. In Enterprise Architect, open the Project Transfer dialog, connect to the .EAP file and the repository, and transfer the project data from the .EAP file to the DBMS repository. <p>You cannot move a model from a source .EAP file of a version earlier than 3.5.0.</p>	<p>Create a Repository^[194]</p> <p>Sparx Systems Scripts</p> <p>Set Up an ODBC Driver^[200]</p> <p>Check Project Data Integrity^[535]</p> <p>Connect to a Data Repository^[209]</p> <p>Perform a Project Data Transfer^[442]</p>

3.3.1 Create a Repository

A repository hosted in a DBMS system requires some initial setup and configuration, and the actual steps required differ depending on the DBMS chosen. In general you must setup your database management system, configure a new database and then populate that database using the scripts provided by Sparx Systems. As a final step, you populate the new repository using a project transfer of the base model.

For more information and the steps involved for specific systems, follow the links below:

Topic	Link
MySQL	Create a MySQL Data Repository ^[195]
Access	Create An Access Data Repository ^[194]
SQL Server	Create a SQL Server Data Repository ^[195]
Oracle	Create an Oracle Data Repository ^[197]
PostgreSQL	Create a PostgreSQL Data Repository ^[197]
Adaptive Server Anywhere	Create an Adaptive Server Anywhere Data Repository ^[198]
MSDE Server	Create an MSDE Server Data Repository ^[198]
Progress OpenEdge	Create a Progress OpenEdge Data Repository ^[199]

Learn more

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)

3.3.1.1 Access 2007 Repository

The process of creating an Access 2007 repository is very simple. Using Access 2007, open the .EAP file and allow Access to convert it to a .ACCDB file. This forms the Access 2007 repository.

Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- If you do not have Access 2007, you can download the Access Database Engine from the [Microsoft](#)

[downloads site](#)

- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

3.3.1.2 MySQL Repository

Creating a MySQL database is beyond the scope of this User Guide - refer to your MySQL documentation for assistance.

Having created your MySQL database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect.

You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

Third Party Tools

If you are unfamiliar with MySQL and DBMS systems in general, you might want to consider a suitable front end tool. MySQL Administrator is one such tool, providing a convenient graphical user interface to enable the creation of databases, execution of scripts, backups and restores.

You might, therefore:

1. Run MySQL Administrator and create a new database
2. Run MySQL Query Browser, and open and execute the MySQL repository script

After creating a MySQL data repository in Enterprise Architect, you must set up the MySQL ODBC drivers.

Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

Learn more

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)
- [MySQL Workbench \(including MySQL Administrator\) download site](#)
- [MySQL ODBC Driver](#)²⁰¹¹

3.3.1.3 SQL Server Repository

Before creating a SQL Server data repository, you must have SQL Server and MDAC 2.6 or higher installed, and access permission to create a new database. Creating a SQL Server database is beyond the scope of this User Guide - refer to your SQL Server documentation for assistance.

Having created your SQL Server database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect. You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)

- Corporate edition Resources page (Trial users)

How to

If you are unfamiliar with SQL Server and DBMS systems in general, you might want to consider a suitable front end tool, such as SQL Enterprise Manager. The following example shows how to create the repository in SQL Enterprise Manager.

Step	Action
1	In SQL Enterprise Manager, locate the server on which to create your new Enterprise Architect model; for example: DBSERVER02\SQLEXPRESS.
2	Right-click and choose the New Database context menu option.
3	Enter a suitable name for the database. Set any file options as required. Ensure that the database collation is set to the alphabet you use, such as Latin or Cyrillic, and case-insensitive.
4	Click on the database to select it, then select the New Query menu option.
5	In the Query window, use the Open File dialog to locate the supplied Enterprise Architect SQL Server Model script file.
6	Click on the Open button. In the drop-down menu, check that you have selected the correct database to run the script in.
7	Click on the Execute button; SQL Server executes the script, which creates the base model for an Enterprise Architect project.

Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn more

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)

3.3.1.4 Oracle Data Repository

Before creating an Oracle data repository, you must have the appropriate version of Oracle (9i, 10g or 11g) and MDAC 2.6 or higher installed, and access permission to create a new database. Creating an Oracle database is beyond the scope of this User Guide - refer to your Oracle documentation for assistance.

Having created your Oracle database, use the SQL script provided by Sparx Systems to create the required table definitions and indexes for Enterprise Architect. You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

Third Party Tools

If you are unfamiliar with Oracle and DBMS systems in general, you might want to consider a suitable front end tool. You could connect to the database and execute the scripts with a program such as Oracle SQL*Plus or SQL Plus Worksheet.

Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

Learn more

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)

3.3.1.5 PostgreSQL Repository

Creating a PostgreSQL database is beyond the scope of this User Guide - refer to your PostgreSQL documentation for assistance.

Having created your PostgreSQL database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect. You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

Third Party Tools

If you are unfamiliar with PostgreSQL and DBMS systems in general, you might want to consider a suitable front end tool. One such tool is *pgAdminIII*, It provides a convenient graphical user interface to enable creation of databases, execution of scripts and restores.

After creating a PostgreSQL data repository in Enterprise Architect, you must set up the PostgreSQL and PostgreSQL ODBC drivers.

Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

Learn more

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)
- [pgAdminIII Download Site](#)
- [Set up a PostgreSQL ODBC Driver](#)^[2021]

3.3.1.6 Adaptive Server Anywhere Repository

Creating an ASA database is beyond the scope of this User Guide - refer to your Sybase Adaptive Server Anywhere documentation for assistance.

Having created your ASA database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect. You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

Third Party Tools

If you are unfamiliar with ASA and DBMS systems in general, you might want to consider a suitable front end tool. Sybase Central is one such tool, that can be installed along with the DBMS. It provides a convenient graphical user interface to enable creation of databases, execution of scripts and restores.

After creating an ASA data repository in Enterprise Architect, you must set up the ASA ODBC driver.

Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

Learn more

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)
- [Setup an Adaptive Server Anywhere ODBC Driver](#)^[2021]

3.3.1.7 MSDE Server Repository

Before creating an MSDE data repository, you must have MSDE Server and MDAC 2.6 or higher installed. Creating an MSDE Server database is beyond the scope of this User Guide - refer to your MSDE Server documentation for assistance.

Having created your MSDE Server database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect. You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

How to

- Create the MSDE Server repository in the same way as a SQL Server repository

Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

Learn more

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)
- [Create an SQL Server Data Repository](#)^[195]

3.3.1.8 Progress OpenEdge Repository

Before creating a Progress OpenEdge data repository, you must have OpenEdge 10.0B03, 10.1B01 or later, and MDAC 2.6 or higher installed, and access permission to create a new database. Creating an OpenEdge database is beyond the scope of this User Guide - refer to your OpenEdge documentation for assistance.

Having created your OpenEdge database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect. You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

How to

Step	Action
1	Run proenv from the OpenEdge menu: Start Programs OpenEdge proenv.
2	Create the database: <code>pr odb <dat abase_name> empty.</code>
3	Start the database server: <code>pr oserve <dat abase_name> - S <port_number ></code>
4	Open Data Administration to add a user: <code>pr owi n32 - db <dat abase_name> - S <port_number > - p _admi n - r x</code>

Step	Action
5	Select Admin Security Edit User List .
6	Close Data Administration.
7	Open the SQL Explorer Tool, connect as <i>sysprogress</i> .
8	Add a user: <pre>create user 'user', 'password'; commit;</pre>
9	Grant privileges: <pre>grant dba, resource to <user>; commit;</pre>

Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

Learn more

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)

3.3.2 Set Up an ODBC Driver for a Connection to a Repository

Setting up an ODBC driver is only necessary for certain database management systems, which require specific set-up in order to operate as a repository.

Other database management systems connect through OLE DB and do not require a driver. However, you can connect to an Oracle repository using either OLE DB or an ODBC driver; if you intend to connect through the ODBC driver, you must first set it up.

For more information and the steps involved for specific systems, follow the links below:

Topic	Link
MySQL	Set up a MySQL ODBC Driver ^[201]
PostgreSQL	Set up a PostgreSQL ODBC Driver ^[202]

Topic	Link
Adaptive Server Anywhere	Set up an Adaptive Server Anywhere ODBC Driver^[205]
Progress OpenEdge	Set up a Progress OpenEdge ODBC Driver^[206]
Oracle	Set up an Oracle ODBC Driver^[208]

Notes

- Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on 64-bit clients, run the 32-bit ODBC Data Source Administrator from C: `Windows\SysWOW64\odbcad32.exe`

3.3.2.1 MySQL ODBC Driver

This topic explains how to set up a MySQL ODBC driver.

Prerequisites:

You must first have installed:

- Microsoft MDAC components
- MySQL DBMS and repository
- MySQL ODBC driver software version 5.1.5 (version 3.51.14 creates problems in incorporating tests in elements)

Use to

- Set up the MySQL ODBC driver so that you can connect to your MySQL repository in Enterprise Architect

How to

Step	Action
1	Select the Windows™ Control Panel Administrative Tools Data Sources (ODBC) option. The ODBC Data Source Administrator window displays.
2	Click on the Add button. The Create New Data Source dialog displays, enabling you to add a new DSN.
3	Select MySQL ODBC 5.1 Driver from the list.

Step	Action
4	Click on the Finish button. The MySQL Connector/ODBC dialog displays.
5	Enter the following configuration details: <ul style="list-style-type: none"> • A data source name for the connection • A description (optional) • The host address of the DBMS server • User name and password • The database name on the selected server
6	To set the advanced options, click on the Details>> button.
7	Select the following checkboxes (where provided): <ul style="list-style-type: none"> • Return matched rows instead of affected rows (Connection or Cursors/Results tab) • Allow big result sets (Connection tab)
8	Click on the Test Connection button to confirm that the details are correct.
9	If the test succeeds, click on the OK button to complete the configuration. If the test does not succeed, review your settings.

Your MySQL connection is now available to use in Enterprise Architect.

Notes

- Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on 64-bit clients, run the 32-bit ODBC Data Source Administrator from C: `Windows\SysWOW64\odbcad32.exe`

Learn more

- [Create MySQL Repository](#) ^[195]
- [Connect To MySQL Data Repository](#) ^[210]

3.3.2.2 PostgreSQL ODBC Driver

This topic explains how to set up a PostgreSQL ODBC driver.

Prerequisites:

You must first have installed:

- Microsoft MDAC components
- PostgreSQL DBMS and repository
- PostgreSQL ODBC driver software version 7.03.01.00 or above (note that versions 8.3.4 and 8.4.1 of the PostgreSQL ODBC Driver are not supported)

Use to

- Set up the PostgreSQL ODBC driver so that you can connect to your PostgreSQL repository in Enterprise Architect

How to

Step	Action
1	Select the Windows™ Control Panel Administrative Tools Data Sources (ODBC) option. The ODBC Data Source Administrator window displays.
2	Click on the Add button. The Create New Data Source dialog displays, enabling you to add a new DSN.
3	Select PostgreSQL UNICODE from the list.
4	Click on the Finish button. The Postgre SQL Connector/ODBC dialog displays.
5	Enter the following configuration details: <ul style="list-style-type: none"> • A data source name for the connection • The actual name of the database • The host address of the DBMS server • User name • A description (optional) • The password
6	To set the advanced options, click on the Datasource button and set the options on Page 1 and Page 2 as shown below:

Step	Action
	<div data-bbox="347 315 1193 1055"> <p>Page 1 Page 2</p> <p><input type="checkbox"/> Disable Genetic Optimizer <input type="checkbox"/> CommLog (C:\psqlodbc_XXXX.log)</p> <p><input checked="" type="checkbox"/> KSQLO(Keyset Query Optimization) <input type="checkbox"/> Parse Statements</p> <p><input checked="" type="checkbox"/> Recognize Unique Indexes <input type="checkbox"/> Cancel as FreeStmt (Exp)</p> <p><input checked="" type="checkbox"/> Use Declare/Fetch <input type="checkbox"/> MyLog (C:\mylog_XXXX.log)</p> <p>Unknown Sizes</p> <p><input checked="" type="radio"/> Maximum <input type="radio"/> Don't Know <input type="radio"/> Longest</p> <p>Data Type Options</p> <p><input checked="" type="checkbox"/> Text as LongVarChar <input checked="" type="checkbox"/> Unknowns as LongVarChar <input type="checkbox"/> Booleans as Char</p> <p>Miscellaneous</p> <p>Max Varchar: <input type="text" value="1024"/> Max LongVarChar: <input type="text" value="1000000"/></p> <p>Cache Size: <input type="text" value="100"/> SysTable Prefixes: <input type="text" value="dd_"/></p> <p>OK Cancel Apply Defaults</p> </div> <div data-bbox="347 1171 1251 1951"> <p>Page 1 Page 2</p> <p><input type="checkbox"/> Read Only <input type="checkbox"/> Row Versioning</p> <p><input type="checkbox"/> Show System Tables <input checked="" type="checkbox"/> Disallow Premature</p> <p><input checked="" type="checkbox"/> LF <-> CR/LF conversion <input type="checkbox"/> True is -1</p> <p><input checked="" type="checkbox"/> Updatable Cursors <input type="checkbox"/> Server side prepare</p> <p><input checked="" type="checkbox"/> bytea as LO</p> <p>Int8 As <input checked="" type="radio"/> default <input type="radio"/> bigint <input type="radio"/> numeric <input type="radio"/> varchar <input type="radio"/> double <input type="radio"/> int4 <input type="text" value="0x0"/> Extra Opts</p> <p>Protocol <input checked="" type="radio"/> 7.4+ <input type="radio"/> 6.4+ <input type="radio"/> 6.3 <input type="radio"/> 6.2 <input type="radio"/> Nop <input type="radio"/> Transaction <input type="radio"/> Statement</p> <p>Level of rollback on errors</p> <p>DID Options <input type="checkbox"/> Show Column <input type="checkbox"/> Fake Index</p> <p>Connect Settings: <input type="text"/></p> <p>OK Cancel Apply</p> </div>

Step	Action
7	If you are using PostgreSQL version 8+, on Page 2 select: <ul style="list-style-type: none"> • The Disallow Premature checkbox • In the Protocol panel, the 7.4+ radio button
8	Click on the OK button to complete the configuration.

Your PostgreSQL connection is now available to use in Enterprise Architect.

Notes

- Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on 64-bit clients, run the 32-bit ODBC Data Source Administrator from C: *Windows\SysWOW64\odbcad32.exe*

Learn more

- [Create a PostgreSQL Repository](#)^[197]
- [Connect To a PostgreSQL Data Repository](#)^[218]

3.3.2.3 ASA ODBC Driver

This topic explains how to set up an Adaptive Server Anywhere (ASA) ODBC driver.

Prerequisites:

You must first have installed:

- Microsoft MDAC components
- Adaptive Server Anywhere - SQL Anywhere Studio 8 or higher, and created a repository
- ASA ODBC driver software (installed with the ASA DBMS)

Use to

- Set up the ASA ODBC driver so that you can connect to your ASA repository in Enterprise Architect

How to

Step	Action
1	Select the Windows™ Control Panel Administrative Tools Data Sources (ODBC) option. The ODBC Data Source Administrator window displays.
2	Click on the Add button.

Step	Action
	The Create New Data Source dialog displays, enabling you to add a new DSN.
3	Select Adaptive Server Anywhere or SQL Anywhere from the list.
4	Click on the Finish button. The ASA Connector/ODBC dialog displays.
5	Enter the following configuration details: <ul style="list-style-type: none"> • A data source name for the connection, on the ODBC tab • User name and password on the Login tab (dba, sql are the defaults when ASA is installed) • The server name and the path to the database, on the Database tab • The network protocol on the Network tab (if the database is on a network location)
6	Return to the ODBC tab and test the connection.
7	Click on the OK button to complete the configuration.

Your ASA connection is now available to use in Enterprise Architect.

Notes

- Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on 64-bit clients, run the 32-bit ODBC Data Source Administrator from C: `Windows\SysWOW64\odbcad32.exe`

Learn more

- [Create an Adaptive Server Anywhere Repository](#)^[198]
- [Connect To an ASA Data Repository](#)^[219]

3.3.2.4 Progress OpenEdge ODBC Driver

This topic explains how to set up a Progress OpenEdge ODBC driver.

Prerequisites:

You must first have installed:

- Microsoft MDAC components
- OpenEdge DBMS and repository
- DataDirect ODBC driver for OpenEdge software (version 4.20 minimum)

Use to

- Set up the Progress OpenEdge ODBC driver so that you can connect to your OpenEdge repository in Enterprise Architect

How to

Step	Action
1	Select the Windows™ Control Panel Administrative Tools Data Sources (ODBC) option. The ODBC Data Source Administrator window displays.
2	Click on the Add button. The Create New Data Source dialog displays, enabling you to add a new DSN.
3	Select DataDirect/OpenEdge SQL Driver from the list.
4	Click on the Finish button. The DSN Configuration dialog displays.
5	Enter the following configuration details: <ul style="list-style-type: none"> • A data source name for the connection • A description (optional) • The host name and port number of the DBMS server • The database name on the selected server • The User ID
6	Click on the Test Connect button to confirm that the details are correct.
7	If the test succeeds, click on the OK button to complete the configuration. If the test does not succeed, review your settings.

Your OpenEdge connection is now available to use in Enterprise Architect.

Notes

- Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on 64-bit clients, run the 32-bit ODBC Data Source Administrator from C:
Windows\SysWOW64\odbcad32.exe

Learn more

- [Create a PostgreSQL Repository](#)^[197]
- [Connect To a PostgreSQL Data Repository](#)^[218]

3.3.2.5 Oracle ODBC Driver

This topic explains how to set up an Oracle ODBC driver.

Prerequisites:

You must first have installed:

- Microsoft MDAC components
- Oracle DBMS and repository

Use to

- Set up the Oracle ODBC driver so that you can connect to your Oracle repository in Enterprise Architect, in preference to using an OLE DB connection

How to

Step	Action
1	Select the Windows™ Control Panel Administrative Tools Data Sources (ODBC) option. The ODBC Data Source Administrator window displays.
2	Click on the Add button. The Create New Data Source dialog displays, enabling you to add a new DSN.
3	Select Oracle in OraDB11g_home1 from the list (or similar, depending on the ODBC installation).
4	Click on the Finish button. The Oracle ODBC Driver Configuration dialog displays.
5	Enter the following configuration details: <ul style="list-style-type: none"> • A data source name for the connection • A description (optional) • The TNS Service Name (click on the drop down arrow and select from the list) • User ID
6	Click on the Test Connection button and enter the Oracle user password to confirm that the details

Step	Action
	are correct.
7	If the test succeeds, click on the OK button to complete the configuration. If the test does not succeed, review your settings.

Your Oracle connection is now available to use in Enterprise Architect.

Notes

- Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on 64-bit clients, run the 32-bit ODBC Data Source Administrator from C: `Windows\SysWOW64\odbcad32.exe`

Learn more

- [Create an Oracle Server Repository](#)^[197]
- [Connect to an Oracle Data Repository \(ODBC\)](#)^[214]

3.3.3 Connect to a Data Repository

This topic describes how to connect to a data repository to:

- Access an Enterprise Architect model in the repository
- Reverse engineer a database schema into a model using the ODBC driver provided by the DBMS vendor

To connect to a repository, you must have the usual **SELECT**, **UPDATE**, **INSERT** and **DELETE** permissions.

For more information and the steps involved for specific systems, follow the links below:

Topic	Link
MySQL	Connect to a MySQL Data Repository ^[210]
SQL Server	Connect To a SQL Server Data Repository ^[212]
Oracle	Connect To an Oracle Data Repository ^[214]
PostgreSQL	Connect To a PostgreSQL Data Repository ^[218]
Adaptive Server Anywhere	Connect To an Adaptive Server Anywhere Data Repository ^[219]

Topic	Link
MSDE Server	Connect To an MSDE Server Data Repository ^[22]
Progress OpenEdge	Connect To a Progress OpenEdge Data Repository ^[22]

Learn more

- [Import Database Schema Using ODBC](#)^[1973]

3.3.3.1 MySQL Data Repository

This topic explains how to connect to a MySQL data repository in Enterprise Architect.

Access **File | Open Project (Ctrl+O)**

Use to

- Connect to the data repository so that you can open a project held in the repository

Prerequisites:

- Create a MySQL Repository
- Set up a MySQL ODBC driver

How to

Step	Action
1	In the Open Project dialog, select the Connect to Server checkbox.
2	Click on the (...) (Browse) button. The Data Link Properties dialog displays.
3	Select Microsoft OLE DB Provider for ODBC Drivers from the list.
4	Click on the Next>> button. The Connection tab displays.
5	Click on the Use data source name radio button and on the drop-down arrow in its field. From the list, select the ODBC driver you have set up to connect to your MySQL repository.

Step	Action
6	If required, type in a User name and Password .
7	If required, type in an initial catalog.
8	Click on the Test Connection button to confirm that the details are correct.
9	If the test does not succeed, revise your settings. If the test succeeds, click on the OK button; the Connection Name & Type dialog displays.
10	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog.
11	If required, select the Encrypt Connection String checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
12	If required, select the Lazy Load checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded. With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.
13	If required, select the Use WAN Optimization checkbox. To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time. If you select this checkbox, complete the next three fields (see your administrator for the correct values); otherwise go to step 17.
14	In the Server field, type the network name or address of the optimizer server.
15	In the Port field, type the port on which the server is running on the remote machine.
16	In the DSN field, type the data source name of the database as it appears on the remote machine.
17	Click on the OK button to complete the configuration.

Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn more

- [Create a MySQL Repository](#)^[195]
- [Set up a MySQL ODBC driver](#)^[201]
- [The WAN Optimizer](#)^[238]
- [Open a Project](#)^[181]

3.3.3.2 SQL Server Data Repository

This topic explains how to connect to a SQL Server data repository in Enterprise Architect.

Access **File | Open Project (Ctrl+O)**

Use to

- Connect to the data repository so that you can open a project held in the repository

Prerequisites:

- Create a SQL Server Repository

How to

Step	Action
1	In the Open Project dialog, select the Connect to Server checkbox.
2	Click on the (...) (Browse) button. The Data Link Properties dialog displays.
3	Select Microsoft OLE DB Provider for SQL Server from the list.
4	Click on the Next>> button. The Connection tab displays.
5	Type in the server details, including Server Name, User Name and Password.
6	Click on the Select the database on the server option and on the drop-down arrow.

Step	Action
	From the list, select the model to connect to.
7	Click on the Test Connection button to confirm that the details are correct.
8	If the test does not succeed, revise your settings. If the test succeeds, click on the OK button; the Connection Name & Type dialog displays.
9	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog.
10	If required, select the Encrypt Connection String checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
11	If required, select the Lazy Load checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded. With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.
12	If required, select the Use WAN Optimization checkbox. To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time. If you select this checkbox, complete the next two fields (see your administrator for the correct values); otherwise go to step 15.
13	In the Server field, type the network name or address of the optimizer server.
14	In the Port field, type the port on which the server is running on the remote machine.
15	Click on the OK button to complete the configuration.

Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn more

- [Create a SQL Server Repository](#)^[195]

- [The WAN Optimizer](#)^[238]
- [Open a Project](#)^[181]

3.3.3.3 Oracle Data Repository (ODBC)

This topic explains how to connect to an Oracle 9i, 10g or 11g data repository in Enterprise Architect using an Oracle ODBC driver.

You can also connect to the Oracle repository using OLE DB.

Access **File | Open Project (Ctrl+O)**

Use to

- Connect to the data repository so that you can open a project held in the repository

Prerequisites:

- Create an Oracle Repository
- Set up an Oracle ODBC driver

How to

Step	Action
1	In the Open Project dialog, select the Connect to Server checkbox.
2	Click on the (...) (Browse) button. The Data Link Properties dialog displays.
3	Select Microsoft OLE DB Provider for ODBC Drivers from the list.
4	Click on the Next>> button. The Connection tab displays.
5	In the Data source field, click on the drop-down arrow and select the data source name.
6	Type in the User name and Password .
7	Click on the Test Connection button to confirm that the details are correct.
8	If the test does not succeed, revise your settings. If the test succeeds, click on the OK button; Oracle prompts you for the password.

Step	Action
9	Type in the password. The Connection Name and Type dialog displays.
10	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog.
11	If required, select the Encrypt Connection String checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
12	If required, select the Lazy Load checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded. With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.
13	If required, select the Use WAN Optimization checkbox. To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time. If you select this checkbox, complete the next two fields (see your administrator for the correct values); otherwise go to step 16.
14	In the Server field, type the network name or address of the optimizer server.
15	In the Port field, type the port on which the server is running on the remote machine.
16	Click on the OK button to complete the configuration.

Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn more

- [Create an Oracle Data Repository](#)^[197]
- [Set up an Oracle ODBC Driver](#)^[208]
- [Connect to an Oracle Data Repository \(OLE DB\)](#)^[216]
- [The WAN Optimizer](#)^[238]

- [Open a Project](#)^[18]

3.3.3.4 Oracle Data Repository (OLE DB)

This topic explains how to connect to an Oracle 9i, 10g or 11g data repository in Enterprise Architect using OLE DB.

You can also connect to the Oracle repository using an Oracle ODBC driver.

Access **File | Open Project (Ctrl+O)**

Use to

- Connect to the data repository so that you can open a project held in the repository

Prerequisites:

- Create an Oracle Repository

How to

Step	Action
1	In the Open Project dialog, select the Connect to Server checkbox.
2	Click on the (...) (Browse) button. The Data Link Properties dialog displays.
3	Select Oracle Provider for OLE DB from the list. Do not select Microsoft OLE DB Provider for Oracle ; Enterprise Architect might not work as expected.
4	Click on the Next>> button. The Connection tab displays.
5	In the Data source field, click on the drop-down arrow and select the data source name (the service name of the Oracle database).
6	Type in the User name and Password .
7	Click on the Test Connection button to confirm that the details are correct.
8	If the test does not succeed, revise your settings.

Step	Action
	If the test succeeds, click on the OK button; the Connection Name and Type dialog displays.
9	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog.
10	If required, select the Encrypt Connection String checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
11	If required, select the Lazy Load checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded. With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.
12	If required, select the Use WAN Optimization checkbox. To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time. If you select this checkbox, complete the next two fields (see your administrator for the correct values); otherwise go to step 15.
13	In the Server field, type the network name or address of the optimizer server.
14	In the Port field, type the port on which the server is running on the remote machine.
15	Click on the OK button to complete the configuration.

Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn more

- [Create an Oracle Data Repository](#)^[19]
- [Connect to an Oracle Data Repository \(ODBC\)](#)^[21]
- [The WAN Optimizer](#)^[23]
- [Open a Project](#)^[18]

3.3.3.5 PostgreSQL Data Repository

This topic explains how to connect to a PostgreSQL data repository in Enterprise Architect.

Access **File | Open Project (Ctrl+O)**

Use to

- Connect to the data repository so that you can open a project held in the repository

Prerequisites:

- Create a PostgreSQL Repository
- Set up a PostgreSQL ODBC driver

How to

Step	Action
1	In the Open Project dialog, select the Connect to Server checkbox.
2	Click on the (...) (Browse) button. The Data Link Properties dialog displays.
3	Select Microsoft OLE DB Provider for ODBC Drivers from the list.
4	Click on the Next>> button. The Connection tab displays.
5	Click on the Use data source name radio button and on the drop-down arrow in its field. From the list, select the ODBC driver you have set up to connect to your PostgreSQL repository.
6	Click on the Test Connection button to confirm that the details are correct.
7	If the test does not succeed, revise your settings. If the test succeeds, click on the OK button; the Connection Name & Type dialog displays.
8	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog.
9	If required, select the Encrypt Connection String checkbox.

Step	Action
	This encrypts and hides the connection details of the database from the users that the connection string is given to.
10	If required, select the Lazy Load checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded. With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.
11	If required, select the Use WAN Optimization checkbox. To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time. If you select this checkbox, complete the next three fields (see your administrator for the correct values); otherwise go to step 15.
12	In the Server field, type the network name or address of the optimizer server.
13	In the Port field, type the port on which the server is running on the remote machine.
14	In the DSN field, type the data source name of the database as it appears on the remote machine.
15	Click on the OK button to complete the configuration.

Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn more

- [Create a PostgreSQL Repository](#)^[197]
- [Set up a PostgreSQL ODBC Driver](#)^[202]
- [The WAN Optimizer](#)^[238]
- [Open a Project](#)^[181]

3.3.3.6 ASA Data Repository

This topic explains how to connect to an Adaptive Server Anywhere data repository in Enterprise Architect.

Access **File | Open Project (Ctrl+O)**

Use to

- Connect to the data repository so that you can open a project held in the repository

Prerequisites:

- Create an ASA Data Repository
- Set up an ASA ODBC driver

How to

Step	Action
1	In the Open Project dialog, select the Connect to Server checkbox.
2	Click on the (...) (Browse) button. The Data Link Properties dialog displays.
3	Select Microsoft OLE DB Provider for ODBC Drivers from the list.
4	Click on the Next>> button. The Connection tab displays.
5	Click on the Use data source name radio button and on the drop-down arrow in its field. From the list, select the ODBC driver you have set up to connect to your ASA repository.
6	Click on the Test Connection button to confirm that the details are correct.
7	If the test does not succeed, revise your settings. If the test succeeds, click on the OK button; the Connection Name & Type dialog displays.
8	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog.
9	If required, select the Encrypt Connection String checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
10	If required, select the Lazy Load checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded. With this setting, the model loads faster and users can begin work sooner, but at the expense of

Step	Action
	later small delays as Enterprise Architect loads specific portions of the model.
11	<p>If required, select the Use WAN Optimization checkbox.</p> <p>To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time.</p> <p>If you select this checkbox, complete the next three fields (see your administrator for the correct values); otherwise go to step 15.</p>
12	In the Server field, type the network name or address of the optimizer server.
13	In the Port field, type the port on which the server is running on the remote machine.
14	In the DSN field, type the data source name of the database as it appears on the remote machine.
15	Click on the OK button to complete the configuration.

Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn more

- [Create an Adaptive Server Anywhere Repository](#)^[198]
- [Set up an ASA ODBC Driver](#)^[205]
- [The WAN Optimizer](#)^[238]
- [Open a Project](#)^[181]

3.3.3.7 MSDE Server Data Repository

To connect to an MSDE Server Data Repository, follow the steps in [Connect to an SQL Server Repository](#)^[212].

Use to

- Connect to the data repository so that you can open a project held in the repository

3.3.3.8 Progress OpenEdge Repository

This topic explains how to connect to a Progress OpenEdge data repository in Enterprise Architect.

Access **File | Open Project (Ctrl+O)**

Use to

- Connect to the data repository so that you can open a project held in the repository

Prerequisites:

- Create a Progress OpenEdge Repository
- Set up a Progress OpenEdge ODBC driver

How to

Step	Action
1	In the Open Project dialog, select the Connect to Server checkbox.
2	Click on the (...) (Browse) button. The Data Link Properties dialog displays.
3	Select Microsoft OLE DB Provider for ODBC Drivers from the list.
4	Click on the Next>> button. The Connection tab displays.
5	Click on the Use data source name radio button and on the drop-down arrow in its field. From the list, select the ODBC driver you have set up to connect to your Progress OpenEdge repository.
6	Enter the User Name and Password .
7	Enter the Initial catalog .
8	Click on the All tab and double-click on Extended Properties .
9	Click on the Connection tab again.
10	In the Property Value field, edit the value to: DefaultSchema=PUB . Click on the Test Connection button to confirm that the details are correct.
11	If the test does not succeed, revise your settings. If the test succeeds, click on the OK button; the Logon to Progress dialog displays.

Step	Action
12	Check the details to ensure that they are correct. Click on the OK button; the Connection Name and Type dialog displays.
13	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog.
14	If required, select the Encrypt Connection String checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
15	If required, select the Lazy Load checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded. With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.
16	If required, select the Use WAN Optimization checkbox. To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time. If you select this checkbox, complete the next three fields (see your administrator for the correct values); otherwise go to step 20.
17	In the Server field, type the network name or address of the optimizer server.
18	In the Port field, type the port on which the server is running on the remote machine.
19	In the DSN field, type the data source name of the database as it appears on the remote machine.
20	Click on the OK button to complete the configuration.

Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn more

- [Create a Progress OpenEdge Repository](#)^[199]
- [Set up a Progress OpenEdge ODBC Driver](#)^[206]
- [The WAN Optimizer](#)^[238]

- [Open a Project](#)^[18]

3.3.4 Upsizing

This topic explains how you upsize your .EAP project or template to one of a range of DBMS repositories.

For more information and the steps involved for specific systems, follow the links below:

Topic	Link
MySQL	Upsize to MySQL ^[22]
Access	Upsize to Access 2007 ^[22]
SQL Server	Upsize to SQL Server ^[22]
Oracle	Upsize to Oracle 9i, 10g or 11g ^[23]
PostgreSQL	Upsize to PostgreSQL ^[23]
Adaptive Server Anywhere	Upsize to Sybase ASA ^[23]
MSDE Server	Upsize to MSDE ^[23]
Progress OpenEdge	Upsize to Progress OpenEdge ^[23]

Notes

- You cannot move a model from a source .EAP file of a version earlier than 3.5.0
- Before proceeding, ensure MDAC 2.6 or higher is installed on your system

3.3.4.1 Upsize to MySQL

This topic explains how to upsize your .EAP project or template for use with MySQL.

The process has the following stages:

- Stage 1: Perform project integrity check
- Stage 2: Install MySQL components
- Stage 3: Select table type
- Stage 4: Create an empty database repository
- Stage 5: Set up the ODBC Driver
- Stage 6: Transfer the data

Perform Project Integrity Check:

Step	Action	See also
1	Open the base project or template .eap file.	
2	Select and run Tools Data Management Project Integrity Check . This ensures your project data is 'clean' before uploading.	Project Integrity Check [535]

Install MySQL Components:

Step	Action	See also
1	Install MySQL version 4.0.3 or higher.	Create a MySQL Repository [195]
2	Install MySQL ODBC 5.1 or higher.	

Select Table Type:

If MyISAM table types are used (default), transactional support is disabled. To enable transactions you must set up MySQL to use InnoDB tables and create the database tables as InnoDB type.

Sparx provide scripts to create both InnoDB- and MyISAM-based repository tables.

Step	Action	See also
1	Set up the <i>MySQL .ini</i> file as required.	
2	If you are using <i>InnoDB</i> tables run the <i>MySQL_InnoDB_EASchema.sql</i> script. If you are using <i>MyISAM</i> tables run the <i>MySQL_MyISAM_EASchema.sql</i> script.	Sparx Systems website

Create Database Repository:

Step	Action	See also
1	Create an empty database repository.	Create a New MySQL

Step	Action	See also
		Repository ^[195]

Set Up the ODBC Driver:

Step	Action	See also
1	<p>Create a suitable ODBC Data Source to point to your new database.</p> <p>Select the following checkboxes:</p> <ul style="list-style-type: none"> • Return matched rows instead of affected rows • Allow big result sets 	Set up a MySQL ODBC Driver ^[201]

Transfer the Project Data:

Step	Action	See also
1	<p>Open Enterprise Architect.</p> <p>Click on the Cancel button on the Open Project screen to open with no project loaded.</p>	
2	<p>Select the Tools Data Management Project Transfer menu option.</p> <p>The Project Transfer dialog displays.</p>	
3	<p>In the Transfer Type panel, select .EAP to DBMS.</p>	
4	<p>In the Source Project field, type the name of the .EAP file to upsize to MySQL.</p>	
5	<p>At the right of the Target Project field, click on the (...) (Browse) button.</p> <p>The Datalink Properties dialog displays.</p>	
6	<p>Select Microsoft OLE DB Provider for ODBC Drivers from the list.</p> <p>Click on the Next button.</p>	
7	<p>In the Use Data source name field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database.</p>	Connect to a MySQL Data Repository ^[210]

Step	Action	See also
	Click on the OK button.	
8	If required, select the Logfile checkbox and type a path and filename for the data transfer log file.	
9	Click on the Transfer button to begin the data transfer process.	

When the process is complete, you have upsized your model to MySQL and can now open it from Enterprise Architect.

3.3.4.2 Upsize to Access 2007

This topic explains how to upsize your .EAP project or template for use with Access 2007.

The process has the following stages:

- Stage 1: Perform project integrity check
- Stage 2: Create an empty database repository
- Stage 3: Transfer the data

Perform Project Integrity Check:

Step	Action	See also
1	Open the base project or template .eap file.	
2	Select and run Tools Data Management Project Integrity Check . This ensures your project data is 'clean' before uploading.	Project Integrity Check ^[535]

Create Database Repository:

Step	Action	See also
1	Create an empty database repository.	Create an Access 2007 Repository ^[194]

Transfer the Project Data:

Step	Action	See also
1	Open Enterprise Architect. Click on the Cancel button on the Open Project screen to open with no project loaded.	
2	Select the Tools Data Management Project Transfer menu option. The Project Transfer dialog displays.	
3	In the Transfer Type panel, select .EAP to DBMS .	
4	In the Source Project field, type the name of the .EAP file to upsize to Access 2007.	
5	At the right of the Target Project field, click on the (...) (Browse) button. The Datalink Properties dialog displays.	
6	Select Microsoft Office 12.0 Access Database Engine OLE DB Provider from the list. Click on the Next button.	
7	On the Data Source Details page of the Connection dialog, type in the full path to the Access 2007 .ACCDB file. Click on the OK button to return to the Project Transfer dialog.	
8	If required, select the Logfile checkbox and type a path and filename for the data transfer log file.	
9	Click on the Transfer button to begin the data transfer process.	

When the process is complete, you have upsized your model to Access 2007 and can now open it from Enterprise Architect.

3.3.4.3 Upsize to SQL Server

This topic explains how to upsize your .EAP project or template for use with SQL Server.

The process has the following stages:

- Stage 1: Perform project integrity check
- Stage 2: Create an empty database repository
- Stage 3: Transfer the data

Perform Project Integrity Check:

Step	Action	See also
1	Open the base project or template .eap file.	
2	Select and run Tools Data Management Project Integrity Check . This ensures your project data is 'clean' before uploading.	Project Integrity Check ^[535]

Create Database Repository:

Step	Action	See also
1	Install SQL Server.	
2	Create the empty database.	Create an SQL Server Repository ^[195]
3	Using a tool such as the SQL Query Analyser, load the <i>SQLServer_EASchema.sql</i> file.	Sparx Systems Website
4	Make sure the new database is the currently active database.	
5	Run the script to create all required data structures.	Create an SQL Server Repository ^[195]

Transfer the Project Data:

Step	Action	See also
1	Open Enterprise Architect. Click on the Cancel button on the Open Project screen to open with no project loaded.	
2	Select the Tools Data Management Project Transfer menu option. The Project Transfer dialog displays.	
3	In the Transfer Type panel, select .EAP to DBMS .	

Step	Action	See also
4	In the Source Project field, type the name of the .EAP file to upsize to SQL Server.	
5	At the right of the Target Project field, click on the (...) (Browse) button. The Datalink Properties dialog displays.	
6	Select Microsoft OLE DB Provider for SQL Server from the list. Click on the Next button.	
7	On the Data Source Details page of the Connection dialog, type in the server name, database name and security details as required. Click on the OK button.	Connect to SQL Server Data Repository ^[212]
8	If required, select the Logfile checkbox and type a path and filename for the data transfer log file.	
9	Click on the Transfer button to begin the data transfer process.	

When the process is complete, you have upsized your model to SQL Server and can now open it from Enterprise Architect.

Notes

- When transferring a project you must have **db_ddladmin** permission in order to execute the **SET IDENTITY_INSERT (table) {ON | OFF}** command

3.3.4.4 Upsize to Oracle 9i, 10g or 11g

This topic explains how to upsize your .EAP project or template for use with Oracle 9i, 10g or 11g.

The process has the following stages:

- Stage 1: Perform project integrity check
- Stage 2: Create an empty database repository
- Stage 3: Transfer the data

Perform Project Integrity Check:

Step	Action	See also
1	Open the base project or template .eap file.	
2	Select and run Tools Data Management Project Integrity Check . This ensures your project data is 'clean' before uploading.	Project Integrity Check ^[535]

Create Database Repository:

Step	Action	See also
1	Install Oracle.	
2	Create the empty database.	Create an Oracle Data Repository ^[197]
3	Using a tool such as the SQL*Plus or SQL Plus Worksheet, load the <i>Oracle_EASchema.sql</i> file.	Sparx Systems Website
4	Make sure the new database is the currently active database.	
5	Run the script to create all required data structures.	Create an Oracle Data Repository ^[197]

Transfer the Project Data:

Step	Action	See also
1	Open Enterprise Architect. Click on the Cancel button on the Open Project screen to open with no project loaded.	
2	Select the Tools Data Management Project Transfer menu option. The Project Transfer dialog displays.	
3	In the Transfer Type panel, select .EAP to DBMS .	
4	In the Source Project field, type the name of the .EAP file to upsize to	

Step	Action	See also
	Oracle.	
5	At the right of the Target Project field, click on the (...) (Browse) button. The Datalink Properties dialog displays.	
6	Select Oracle Provider for OLE DB from the list. Click on the Next button.	
7	On the Connection page of the Datalink Properties dialog, enter the Oracle service name in the Data Source field, and the user name and password as required. Click on the OK button.	Connect to an Oracle Data Repository ^[214]
8	If required, select the Logfile checkbox and type a path and filename for the data transfer log file.	
9	Click on the Transfer button to begin the data transfer process.	

When the process is complete, you have upsized your model to Oracle and can now open it from Enterprise Architect.

Notes

- When transferring a project to Oracle you must have permission to execute the **CREATE SEQUENCE** command

3.3.4.5 Upsize to PostgreSQL

This topic explains how to upsize your .EAP project or template for use with PostgreSQL.

The process has the following stages:

- Stage 1: Perform project integrity check
- Stage 2: Create an empty database repository
- Stage 3: Install PostgreSQL Driver
- Stage 4: Transfer the data

Perform Project Integrity Check:

Step	Action	See also
1	Open the base project or template .eap file.	
2	Select and run Tools Data Management Project Integrity Check . This ensures your project data is 'clean' before uploading.	Project Integrity Check ^[535]

Create Database Repository:

Step	Action	See also
1	Install PostgreSQL, version 7.3.2 or higher.	
2	Create the empty database.	Create a PostgreSQL Repository ^[197]
3	From the PSQL command line, or using a tool such as the PostgreSQL command line, pgAdminIII or EMS PostgreSQL Manager, load the <i>PostgreSQL_EASchema.sql</i> file.	Sparx Systems Website
4	Run the script to create all required data structures.	Create a PostgreSQL Repository ^[197]

Set Up the ODBC Driver:

Step	Action	See also
1	Install psqLODBC, version 7.03.01.00 or higher (but do not use version 8.4.1).	
2	Create a suitable ODBC Data Source to point to your new database.	Set up a PostgreSQL ODBC Driver ^[202]

Transfer the Project Data:

Step	Action	See also
1	Open Enterprise Architect. Click on the Cancel button on the Open Project screen to open with no project	

Step	Action	See also
	loaded.	
2	Select the Tools Data Management Project Transfer menu option. The Project Transfer dialog displays.	
3	In the Transfer Type panel, select .EAP to DBMS .	
4	In the Source Project field, type the name of the .EAP file to upsize to SQL Server.	
5	At the right of the Target Project field, click on the (...) (Browse) button. The Datalink Properties dialog displays.	
6	Select Microsoft OLE DB Provider for ODBC Drivers from the list. Click on the Next button.	
7	On the Use Data Source Name field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database. Click on the OK button.	Connect to a PostgreSQL Data Repository ^[218]
8	If required, select the Logfile checkbox and type a path and filename for the data transfer log file.	
9	Click on the Transfer button to begin the data transfer process.	

When the process is complete, you have upsized your model to PostgreSQL and can now open it from Enterprise Architect.

Notes

- During the transfer, if an error message displays reporting '*...nonstandard use of \ in a string literal...*', set the server variable in the *postgresql.conf* file to: *escape_string_warning = off*

3.3.4.6 Upsize to Sybase ASA

This topic explains how to upsize your .EAP project or template for use with Sybase Adaptive Server Anywhere.

The process has the following stages:

- Stage 1: Perform project integrity check

- Stage 2: Create an empty database repository
- Stage 3: Transfer the data

Perform Project Integrity Check:

Step	Action	See also
1	Open the base project or template .eap file.	
2	Select and run Tools Data Management Project Integrity Check . This ensures your project data is 'clean' before uploading.	Project Integrity Check ^[535]

Create Database Repository:

Step	Action	See also
1	Install Adaptive Server Anywhere - SQL Anywhere Studio 8 or higher. This also installs the ASA ODBC driver.	
2	Using Sybase Central, create a new database for the Enterprise Architect repository.	Create an Adaptive Server Anywhere Repository ^[198]
3	Open Interactive SQL and load the <i>ASA_EASchema.sql</i> file.	Sparx Systems Website
4	Run the script to create all required data structures.	Create an Adaptive Server Anywhere Repository ^[198]
5	Create a suitable ODBC Data Source to point to your new database.	Set up an Adaptive Server Anywhere ODBC Driver ^[205]

Transfer the Project Data:

Step	Action	See also
1	Open Enterprise Architect. Click on the Cancel button on the Open Project screen to open with no project loaded.	

Step	Action	See also
2	Select the Tools Data Management Project Transfer menu option. The Project Transfer dialog displays.	
3	In the Transfer Type panel, select .EAP to DBMS .	
4	In the Source Project field, type the name of the .EAP file to upsize to ASA.	
5	At the right of the Target Project field, click on the (...) (Browse) button. The Datalink Properties dialog displays.	
6	Select Microsoft OLE DB Provider for ODBC Drivers from the list. Click on the Next button.	
7	In the Use Data source name field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database. Click on the OK button.	Connect to an Adaptive Server Anywhere Data Repository [219]
8	If required, select the Logfile checkbox and type a path and filename for the data transfer log file.	
9	Click on the Transfer button to begin the data transfer process.	

When the process is complete, you have upsized your model to Adaptive Server Anywhere and can now open it from Enterprise Architect.

3.3.4.7 Upsize to MSDE

To upsize your model to MSDE, follow the steps in [Upsize to SQL Server](#) [228].

3.3.4.8 Upsize to Progress OpenEdge

This topic explains how to upsize your .EAP project or template for use with Progress OpenEdge.

The process has the following stages:

- Stage 1: Perform project integrity check
- Stage 2: Create an empty database repository
- Stage 3: Transfer the data

Perform Project Integrity Check:

Step	Action	See also
1	Open the base project or template .eap file.	
2	Select and run Tools Data Management Project Integrity Check . This ensures your project data is 'clean' before uploading.	Project Integrity Check ^[535]

Create Database Repository:

Step	Action	See also
1	Install OpenEdge, version 10.0B3 or higher.	
2	Create an empty OpenEdge database for the new repository.	Create a Progress OpenEdge Repository ^[199]
3	Download the <i>OpenEdge_EASchema.sql</i> file from the Sparx Systems Registered Users Resources page.	Sparx Systems Website
4	Make sure the new database is selected as the current database.	
5	Run the script to create all required data structures.	Create a Progress OpenEdge Repository ^[199]
6	Install OpenEdge ODBC Driver, 10.0B or higher. Create a suitable ODBC Data Source to point to your new database.	Set Up a Progress OpenEdge ODBC Driver ^[206]

Transfer the Project Data:

Step	Action	See also
1	Open Enterprise Architect. Click on the Cancel button on the Open Project screen to open with no project loaded.	

Step	Action	See also
2	Select the Tools Data Management Project Transfer menu option. The Project Transfer dialog displays.	
3	In the Transfer Type panel, select .EAP to DBMS .	
4	In the Source Project field, type the name of the .EAP file to upsize to Progress OpenEdge.	
5	At the right of the Target Project field, click on the (...) (Browse) button. The Datalink Properties dialog displays.	
6	Select Microsoft OLE DB Provider for ODBC Drivers from the list. Click on the Next button.	
7	In the Use Data source name field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database. Click on the OK button.	Connect to a Progress OpenEdge Data Repository ↑
8	If required, select the Logfile checkbox and type a path and filename for the data transfer log file.	
9	Click on the Transfer button to begin the data transfer process.	

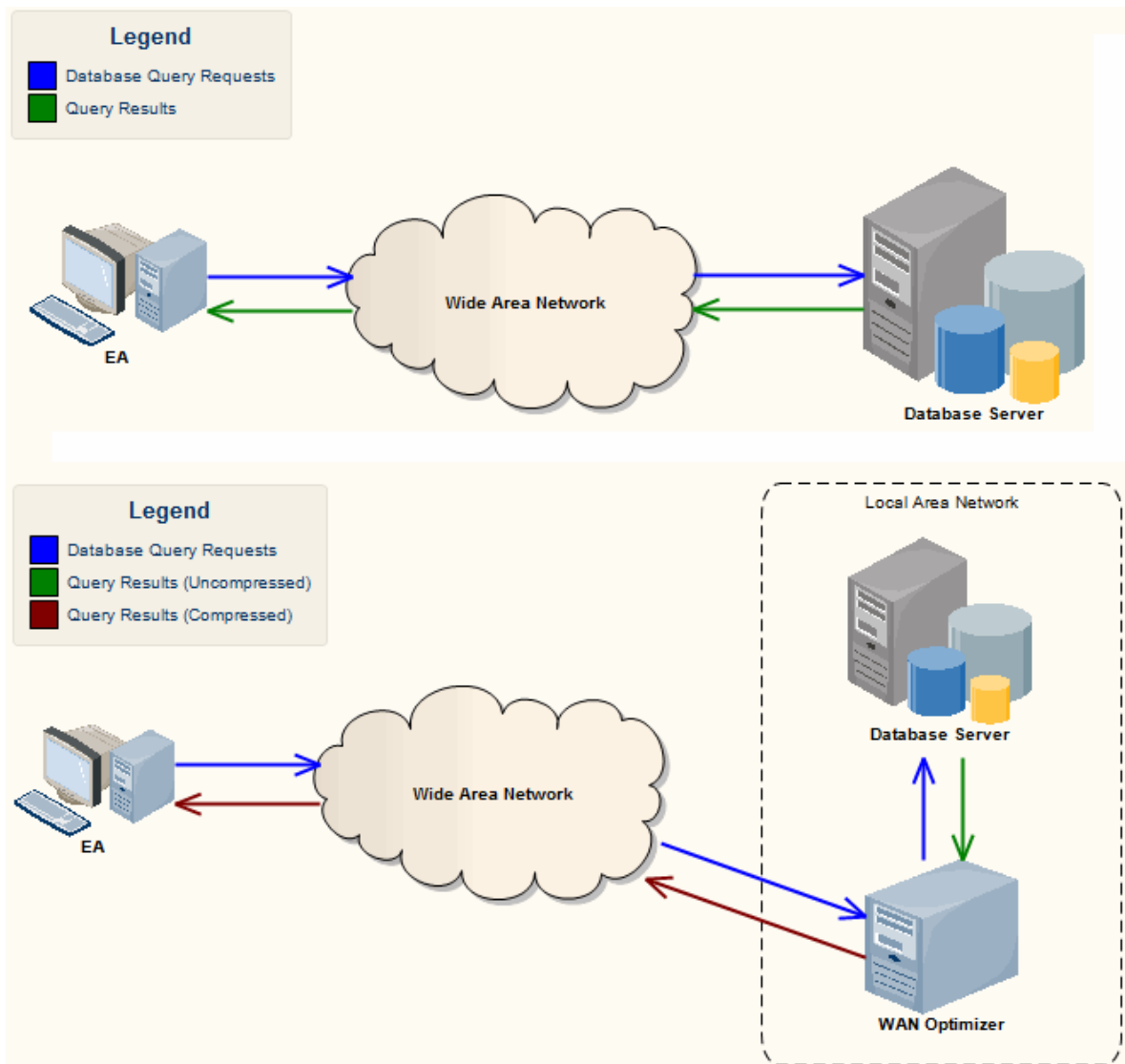
When the process is complete, you have upsized your model to Progress OpenEdge and can now open it from Enterprise Architect.

3.3.5 The WAN Optimizer

The Sparx *Wide Area Network (WAN) Optimizer* is a lightweight server installed on a Local Area Network (LAN) connection to a Database Management System (DBMS) that hosts an Enterprise Architect repository. You can configure the server to listen for client connections on a particular port; it acts as a local proxy to execute queries and return the results in a compressed format to the client.

The WAN Optimizer significantly improves Enterprise Architect's performance in a WAN by reducing the amount of data transmitted and, in turn, the number of network calls made.

In the following diagram, transmission between Enterprise Architect and a DBMS is depicted first without and then with the WAN Optimizer.



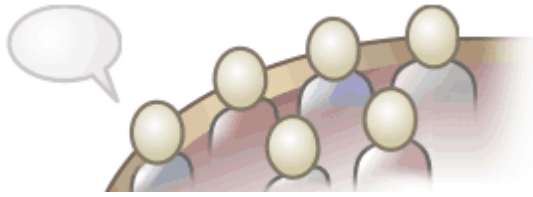
You can download the WAN Optimizer installer from the Downloads page of the Registered Users section of the Sparx Systems website. The Wan Optimizer Service installer package provides two installable features for the target machine:

- WAN Optimizer Service - the installer also helps register and start the service on the target machine, and add it to the Windows Startup folder
- WAN Optimizer Admin Client - to enable an administrator to administer and configure the service from a remote client.

The Optimizer has its own *Sparx WAN Optimizer User Guide*. See that User Guide for more information on:

- WAN Optimizer Components
- Installing and Starting the WAN Optimizer Service
- Configuring the Service
- Troubleshooting

3.4 Team Development



Enterprise Architect enables you to develop a project across a team of people, so that each person can access the latest data without the risk of damaging or losing that data.

Topics

Topic	Detail	See also
Making project data available in a distributed environment	Enterprise Architect offers a diverse set of functionality designed specifically for sharing projects in team-based and distributed development environments; for example: network deployment of model repositories, replication and XML Import/Export.	Project Sharing ^[240]
Applying security to the model	Enterprise Architect Security is a means of improving collaborative design and development by preventing concurrent editing, and limiting the possibility of inadvertent model changes by users not designated as model authors.	Configure User Security ^[251]
Using an internal discussion forum	The Enterprise Architect Project Team Review facility is a discussion forum within your development team community to discuss the development and progress of a project.	Team Review Tools ^[279]
Building company policy and project development guidelines into the project	Enterprise Architect enables you to create workflow scripts that provide a robust approach to applying company policy and strengthening project development guidelines, by validating against the policy and procedures within the model itself.	Workflow Scripts ^[298]
Standardizing and re-using project data	You can import and export Reference data (including Glossary and Issue information) from .XML files of another iteration of the same model, or of a different model.	Sharing Reference Data ^[305]

3.4.1 Project Sharing

Enterprise Architect offers a diverse set of functionality designed specifically for sharing projects in team-based and distributed development environments, through network deployment of model repositories, replication and XML Import/Export.

Topics

Topic	Detail	See also
Network Deployment	<p>Network deployment is possible under two different schemas:</p> <ul style="list-style-type: none"> • .EAP based repositories or • DBMS server based repositories <p>DBMS server based repositories offer better:</p> <ul style="list-style-type: none"> • Response times than .EAP files on networks, due to the inherent structure of the DBMS • Solutions when networking problems are encountered, as they have the ability to backtrack transactions caused by external breakdowns 	Share a Project [242]
Replication	<p>Replication enables data interchange between .EAP based repositories and is suitable for where many different users work independently in parallel development.</p> <p>Modelers merge their changes into a Design Master only as required; it is recommended that a backup is carried out prior to replication.</p> <p>Replication cannot be performed on repositories stored on a DBMS server.</p>	Replication [244]
XMI Import and Export	<p>XMI Import/Export can be used to export and share discrete packages between developers; XMI enables the export of packages into XML files which can then be imported into any model.</p> <p>Package control can be used to set up packages for version control and to enable batch export of packages using XMI; Version Control enables a repository to be maintained by a third-party source code control application that is used to control access and record revisions.</p>	XMI Import/Export [411] Version Control [312] Controlled Packages [422]

Notes

- DBMS Repository support is available with the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect

Learn more

- [Deployment of Enterprise Architect white paper](#)

3.4.1.1 Share Enterprise Architect Projects

The most efficient way of using Enterprise Architect to manage a team development is to *share* a project amongst a team of designers, developers and analysts.

Guide:

Topic	Detail	See also
By Sharing a project	<ul style="list-style-type: none"> • Many people can work on the model at the same time and contribute their particular skill • Team members can always see what the latest changes are, keeping the team informed and up to date with the project status 	Refresh View of Shared Project ^[242]
You can share a project in three ways	<ul style="list-style-type: none"> • Place the project in a shared network directory • Use replication • Use a shared DBMS-based repository 	Share Projects on Network Drive ^[244] Replication ^[244] Server Based Repositories ^[193]

Notes

- Project Sharing and Replication are enabled in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- DBMS repositories are supported in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect

3.4.1.1.1 Refresh View of Shared Project

When a user of a shared model checks out a package and makes changes, other users can see those changes by refreshing their view of the package or the changed diagram within the package, in a number of ways.

Refresh	Detail	See also
Project	<ul style="list-style-type: none"> • Right-click on the package name in the Project Browser and select the Contents Reload Current Package context menu option • Select the File Reload Current Project menu option • Select the Reload Project icon in the Project toolbar • Press (Ctrl+Shift+F11) • Close the project and reopen it 	Contents Submenu ^[577] File ^[927] Project Toolbar ^[147]

Refresh	Detail	See also
Diagram	<ul style="list-style-type: none"> Select the Window Reload Current View menu option Right-click on the opened diagram tab in the diagram view, and select the Reload <diagram name> context menu option 	Window Menu ^[129] Diagram Tabs ^[695]

3.4.1.2 Share Projects on Network Drive

The easiest way to share a project amongst a work group of developers and analysts is to place the project file on a shared network drive, to which people connect from their workstations.

Individual developers and analysts can then open and work on the project concurrently.

Network Issues	See also
<p>Enterprise Architect accepts a number of concurrent connections without issue; however:</p> <ul style="list-style-type: none"> There can be occasional 'lock-outs' when one user tries to access or update something another user is in the process of modifying Changes to the Project Browser (and other project views) are not automatically updated; to compensate for this, users must occasionally reload their project to see changes made by other users If two or more people work on the same diagram concurrently, unexpected results can occur; it is best to allow only one analyst to work on a diagram at a time (.EAP files only) If a user's machine crashes, the network suffers an outage or a machine is turned off unexpectedly, the project file might require repair to compensate for the sudden inconsistency; a repair facility is provided (Tools Data Management Manage .EAP File Repair .EAP File) to carry out this task 	Refresh Model View ^[242] Repair a Project ^[540]

3.4.1.3 Distributed Development

Enterprise Architect supports distributed development using two different techniques.

Topic	Detail	See also
Replication	Enables geographically separated analysts to update and modify parts of the model in replicas, then merge these back together at a central location.	Replication ^[244]
XMI Import/Export	Use XMI-based import/export to export discrete packages to XML to share among the development team; this has several	Import/Export ^[417]

Topic	Detail	See also
	<p>benefits over replication:</p> <ul style="list-style-type: none"> • You can assemble a model from only the parts necessary to get your job done • You can assemble a full model if required • You can assemble a model from different package versions for different purposes (such as customer visible, internal release only) • You can roll-back parts of a model as required • There is less chance of 'collisions' between developers if each works on a discrete package • The process is controllable using a version control system, or through package control <p>XMI based import/export is UML1.3 / XMI1.1 compliant; you can also write XML based tools to manipulate and extract information from XML files to enhance the development process.</p> <p>XMI-based Import/Export is accessed through Project Model Import/Export.</p>	<p>Version Control^[312]</p> <p>Controlled Packages^[422]</p> <p>XMI Import Export Options^[102]</p>

3.4.1.4 Replication

Apart from sharing Enterprise Architect projects in real time over a network, you can also share projects using replication.

Guide:

Topic	Detail	See also
Replication	<p>Replication is a powerful means of sharing projects between isolated or mobile users; it enables different users to work independently of one another, and to merge their changes at a later time.</p> <p>In Replication:</p> <ul style="list-style-type: none"> • A project is converted to a design master, then replicas are made of the master • Users take the replicas away, modify the project, then bring their replicas back to be synchronized with the master file 	
Use Replication	To use replication	Design Master ^[246]

Topic	Detail	See also
	<ol style="list-style-type: none"> 1. Convert the base project to a design master. 2. Create replicas from the design master. 3. Take the replica away and work on it as required, then bring it back for synchronization with the design master. 4. Synchronize the replicas, during which all changes to both the master and the replica are propagated in both directions, so they both finally contain the same information. 	Create Replicas ^[246] Synchronize Replicas ^[247]
Enterprise Architect Merge Rules	<p>Enterprise Architect follows these rules in merging:</p> <ul style="list-style-type: none"> • Additions are cumulative; that is, two replicas each creating three new Classes result in six new Classes after merging • Deletions prevail over modifications; if one replica changes a Class name and other deletes the Class, merging replicas results in both files losing the Class <p>Conflicting modifications appear in the Resolve Replication Conflicts dialog.</p>	Resolve Conflicts ^[250]
Upgrades and Replicas	<p>When you upgrade your version of Enterprise Architect, you must not open a replica until you have opened the design master and then synchronized the replicas with the master; you cannot directly upgrade a replica.</p>	Upgrade Replicas ^[249]
Avoid Change Collisions	<p>If two or more people make changes to the same element, Enterprise Architect arbitrarily overwrites one person's change with the other's; to avoid this, different users should work on different packages.</p> <p>However, since Enterprise Architect does not enforce this rule, it is possible for users' work to conflict; to minimize difficulties, please note the following guidelines:</p> <ul style="list-style-type: none"> • If users are likely to have worked in the same area of the model, they should both witness the synchronization and confirm that they are happy with the net result • If small pieces of information have been lost, they should be typed into one of the merged models after synchronization • If a large piece of information has been lost (for example, an overwritten large Class note) use the 	Resolve Conflicts ^[250]

Topic	Detail	See also
	Resolve Replication Conflicts dialog	
Disable or Remove Replication Features	If you have converted a project to a design master but now want to disable the replication features, you can remove Replication; ensure that you back up all your files first.	Remove Replication ^[245]

3.4.1.4.1 Design Masters

A design master is the first converted Enterprise Architect project that supports replication.

Access [Tools | Data Management | Manage .EAP File | Make Design Master](#)

Use to

- Create the master project from which you create replicas that can be modified independently of the master project and re-merged later

How to

To create a design master

Step	Action
1	Take a back-up of the required Enterprise Architect project.
2	Select the project in the Project Browser.
3	Select the Make Design Master menu option and follow the on-screen instructions.

Learn more

- [Create Replicas](#) ^[246]

3.4.1.4.2 Create Replicas

A replica is one of several copies of the design master of an Enterprise Architect project.

Access [Tools | Data Management | Manage .EAP File | Create New Replica](#)

Use to

- Create a copy of the master project for you or another user to modify independently and re-merge later

How to

To create a replica

Step	Action
1	First create a design master, then select the Create New Replica menu option and follow the on-screen instructions.
2	Edit the replica over time and, when required, return the file for merging with the design master.

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Replicas** permission to create a replica

Learn more

- [Design Masters](#) ^[246]
- [Manage Replicas User Permissions](#) ^[264]

3.4.1.4.3 Synchronize Replicas

Synchronizing replicas combines the changes made to each file.

Access [Tools](#) | [Data Management](#) | [Manage .EAP File](#) | [Synchronize Replicas](#)

Use to

- Merge the changes made to each replica with the design master, so that a new set of replicas with all changes can be generated and distributed
- Combine the changes made to two replicas, should it be necessary for two team members to combine their work

How to

To synchronize replicas

Step	Action
1	Open the design master project file (or the first required replica).
2	Select the Synchronize Replicas menu option.

Step	Action
3	Locate and select the (second) required replica to merge the open project and the replica.

Notes

- Information is copied both ways, including deletes, updates and inserts; both projects end up containing identical information
- If this process generates 'conflicting changes' errors, you should review and, if necessary, resolve these conflicts
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Replicas** permission to create a replica

Learn more

- [Resolve Conflicts](#)^[250]
- [Manage Replicas User Permissions](#)^[264]

3.4.1.4.4 Remove Replication

Replication makes many changes to the database structure of your model, so the model file becomes considerably larger with additional information; you might, therefore, decide not to use the replication feature any more.

Access **Tools | Data Management | Manage .EAP File | Remove Replication**

Use to

- Remove replication from your model

How to

To remove replication from your model

Step	Action
1	Ensure that you have open a temporary repository (not the one having replication removed) - the menu option is not available if no repository is open.
2	Select the Remove Replication menu option. The Remove Replication Wizard displays.

Step	Action
3	Enter the full path and file name of the project to have replication removed. Click on the Next button.
4	Enter the full path and file name of the base Enterprise Architect model (with no replication) to act as template. Click on the Next button.
5	Enter the full path and required file name for the output file. Click on the Next button.
6	Select whether to have a log file created and, if so, enter a file name for the log file.
7	Click on the Run button to begin removing replication. Enterprise Architect creates a new project containing all the model information. Your model has now had replication removed, and should be considerably smaller.

Notes

- You cannot remove replication from a model with Auditing enabled - if you want to remove replication:
 - Disable Auditing
 - If prompted to do so, allow Enterprise Architect to roll back the database version
 - Remove replication

Learn more

- [Auditing](#)^[382]
- [Auditing Settings](#)^[384]

3.4.1.4.5 Upgrade Replicas

A new release of Enterprise Architect might contain changes to the underlying project structure, such as more tables or changed queries. If you use Replication, you must take care with your upgrade.

Guide:

Topic	Detail	See also
Open Design Master Project first	After installing the new version of Enterprise Architect, it is very important that you open the <i>design master</i> before opening any of the replicas with the updated version.	Design Masters ^[246]

Topic	Detail	See also
Changes to the database design	Changes to the database design in a replicated project can ONLY be done to the design master; trying to update a replica at best does nothing, and at worst causes the update of the master to fail.	
Propagate Changes	Design changes are propagated through to the replicas the next time the replicas are synchronized with the master.	Synchronize Replicas [247]
Alternatively	One other strategy is to remove replication from a copy of the replica set, upgrade that project and convert it into a new design master from which new replicas are created.	Remove Replication [248]

3.4.1.4.6 Resolve Conflicts

If two or more people each work on the same model object in their respective replicas between synchronizations, the replication engine has problems in resolving which change is the master.

Guide:

Topic	Detail	See also
Avoid the problem	Ensure that each team member always works in a separate area of the model within their replica.	
Check for conflicts	After synchronizing replicas, open the Resolve Conflicts dialog (see below this table) and check if there were any conflicts.	
Response to conflicts	<p>When a project record has been modified in different ways by different users, the replication engine selects one of the conflicting values based on rules within the JET replication manager.</p> <p>However, the replication engine stores the discarded changes and flags the conflict on the Resolve Conflicts dialog so that you can choose to roll in the discarded change instead.</p> <p>Normally it is not necessary or desirable to examine conflicts, since they represent relatively inconsequential pieces of information that can very easily be modified through the normal Enterprise Architect interface; for example, by moving a diagram element.</p> <p>The only case in which the Resolve Conflicts dialog should be used is where a substantial piece of information has been overridden by a user, and you want to retrieve it .</p>	

Access [Tools | Data Management | Manage .EAP File | Resolve Replication Conflicts](#)

Use to

- Select which of two conflicting changes you should save to the Design Master and/or replicas, where a substantial piece of information has been overridden by a user and you want to retrieve it

How to

To resolve conflicts in changes to the same model object in two separate replicas

Step	Action
1	Synchronize a replica with the Design Master, and display the Resolve Conflicts dialog.
2	In the Table with Conflicts list, click on the entry that is likely to contain the lost information.
3	Click on each entry in the Conflicting Records list. When the lost information appears in the Conflict Details list, click on the Overwrite with Conflict button.
4	In the Conflicting Records list, you can also copy each Row ID to the clipboard by right-clicking on the row and selecting the Copy to Clipboard menu option. You can locate the object that is in conflict in the Project Browser or diagrams, by right-clicking on it and selecting the Find in Project Browser or Find in Diagrams context menu option.

3.4.2 Configure User Security

User security in Enterprise Architect can be used to limit the access to update functions within the model.

Topics

Topic	Detail	See also
User Security	<p>Security in Enterprise Architect is a means of improving collaborative design and development by preventing concurrent editing and limiting the possibility of inadvertent model changes by users not designated as model authors.</p> <p>User security is not enabled by default in Enterprise Architect; the Administrator must enable it first.</p> <p>Where user security is enabled a password is required to log in to the model.</p> <p>Elements can be locked per user or per group.</p>	Enable Security ^[253]
Security Policies	<p>Enterprise Architect User Security offers two policies:</p> <ul style="list-style-type: none"> • Standard security mode - all elements are initially 	Security Policy ^[255]

Topic	Detail	See also
	<p>unlocked and, as necessary, a user can set a user or group lock on any element or set of elements in order to make changes and protect those changes</p> <ul style="list-style-type: none"> • Rigorous security mode - an Enterprise Architect model is read-only and nothing in the model can be edited until explicitly checked out with a user lock 	
Security Operations For Administrators	<p>A number of security tasks can only be performed by users with Administrative rights to the model.</p>	<p>Set Security Policy ^[255]</p> <p>Enable Security ^[253]</p> <p>Maintain Users ^[256]</p> <p>Import User IDs From Active Directory ^[257]</p> <p>Change User Passwords ^[270]</p> <p>Assign User To Groups ^[260]</p> <p>View All User Permissions ^[261]</p> <p>Maintain Groups ^[262]</p> <p>View and Manage Locks ^[268]</p> <p>Password Encryption ^[269]</p> <p>Create Workflow Scripts ^[298]</p>
Security Operations For Users	<p>Other Security tasks can be performed by users who do not have Administrative rights.</p>	<p>Lock Model Elements ^[271]</p> <p>Lock Packages ^[273]</p> <p>Apply a User Lock ^[274]</p> <p>Identify Who Has Locked An Object ^[277]</p> <p>Locked Element Indicators ^[276]</p> <p>Manage Your Own Locks ^[278]</p> <p>Change Your Own Password ^[270]</p>
Security Scripts	<p>With workflow administration permissions, you can also develop workflow scripts using the Scripting window.</p>	<p>List of Available Permissions ^[264]</p>

Topic	Detail	See also
	Workflow scripts validate and control user input.	Workflow Scripts ^[298] Scripting ^[2372] Workflow Script Functions ^[299]

Notes

- User security can be enabled in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- For a number of operations in Enterprise Architect, if security is enabled a user must have the appropriate user or group access permission to perform the operation; however, if security is not enabled, the user does not have to have these access permissions

3.4.2.1 Enable Security

User security is not enabled by default in Enterprise Architect, the Administrator must enable it using an Authorization key obtained from the Sparx Systems website.

Access [Project | Security | Enable Security](#)

Use to

- Enable User Security on a project
- Disable User Security, in exactly the same way

How to

Step	Action	See also
1	Obtain the Authorization Key from the Sparx Systems website on: <ul style="list-style-type: none"> • the Team Modeling Resources section (Trial User) or • the Registered Users section (Registered User; you must also have your Registered Users login and password) 	Team Modeling Resources web pages Registered Users web pages
2	In Enterprise Architect, select the Enable Security menu option. The Enter authorization dialog displays.	
3	In the Enter authorization key field, type the authorization key from the Sparx Systems website.	
4	If required, select the Automatically apply Exclusive Edit Locks to	Security Policy ^[255]

Step	Action	See also
	<p>diagrams checkbox.</p> <p>This option is ignored in <i>Require User Lock</i> security mode.</p> <p>In standard (<i>User/Group Locking</i>) mode, this option blocks multiple users from simultaneously attempting to modify the same diagram (see <i>Notes</i> below).</p>	
5	<p>Click on the OK button.</p> <p>Security is enabled, and an Admin user and user group are created with all access rights and a password of password.</p>	<p>List of Available Permissions ^[264]</p>
6	<p>Select the Project Security Login as Another User menu option, and log in as Admin with the initial password of password.</p> <p>If required, you can change the Admin password.</p>	<p>Change Password ^[270]</p>
7	<p>Set up users and permissions as required.</p>	<p>Maintain Users ^[256]</p> <p>Assign User To Groups ^[260]</p> <p>Set Up Single Permissions ^[260]</p> <p>Maintain Groups ^[262]</p> <p>Set Group Permissions ^[263]</p>

Notes

- Once security has been enabled, you must have the **Security - Enable/Disable** access right to turn it off - the initial administrator automatically has this access right; the system prompts you to log off the project and log on again, but this is not strictly necessary
- If you re-enable security, be aware that any changes you have made to the **Admin** user (password and reduced access rights) are reset to password and full access
- The **Automatically apply Exclusive Edit Locks to diagrams** option is not displayed when disabling security, therefore to toggle the setting whilst security is enabled you must disable security and re-enable it; security settings (users, groups and permissions) and locks on elements, are NOT affected by this action
- If the **Automatically apply Exclusive Edit Locks to diagrams** option is selected, as a user modifies a diagram Enterprise Architect automatically applies a User Lock to the diagram, preventing any other user from modifying it
 It is creating difference between the database and buffer versions of the diagram that triggers the temporary lock, and elimination of difference that releases the lock; therefore, Enterprise Architect releases the lock when:

- The user saves the changes to the diagram, with the **Save** icon or keyboard keys
- The user undoes the last remaining action in the **Undo** list
- The user saves or discards changes via the system prompt when they close the diagram

If the diagram already has a User Lock or Group Lock that does not exclude the current user, this lock is set aside and saved when the temporary User Lock is applied. When the temporary User Lock is released, the pre-existing lock is restored

3.4.2.2 Security Policy

There are two possible security policies in Enterprise Architect.

Guide:

Topic	Detail	See also
User/Group Locking mode	<p>All elements and diagrams are considered unlocked and anyone can edit any part of the project.</p> <p>However, when you edit a diagram, package or element, you lock the element or set of elements at either the user level or group level.</p> <p>This mode is good for cooperative work groups where there is a solid understanding of who is working on which part of the model, and locking is used mainly to prevent further changes or to limit who has access to a part of the model.</p>	
Require User Lock to edit mode	<p>More rigorous: the Enterprise Architect model is read-only, and everything is locked so that nobody can edit anything unless they explicitly check out the object with a user lock.</p> <p>A single 'check out' function operates on a diagram to check out the diagram and all contained elements in one go.</p> <p>There are also functions on the context (right-click) menus of packages, diagrams and elements in the Project Browser to apply a user lock when this mode is in use.</p> <p>You would use this mode when there is a strict requirement to ensure only one person can edit a resource at one time; this is suitable for much larger projects where there might be less communication between users.</p>	

Notes

- Toggle between the security modes using **Project | Security | Require User Lock to Edit** - deselected for **User/Group Locking** mode, selected for **Require User Lock to edit** mode
- When you add new elements in **User/Group Locking** mode (elements editable by default), no user lock is created automatically for the newly created element
- When you add new elements in **Require User Lock to Edit** mode (elements locked by default), a user lock is created on the new element to enable instant editing

3.4.2.3 Maintain Users

If you enable security you have access to the Security Users dialog, which you can use to set up more users for your model.

Access [Project | Security | Manage Users](#)

Use to

- Define the user IDs and initial passwords for the users that are to have access to the model

How to

To set up a user for your model

Step	Action	See also
1	<p>Select the Manage Users menu option.</p> <p>The Security Users dialog displays, which you can use to set up new users by providing their name and other details</p> <p>You can also:</p> <ul style="list-style-type: none"> • Import User IDs from a Windows Active Directory • Assign User IDs to User Groups • Set up single permissions for a user • View all permissions for the currently selected user 	<p>Import User IDs from a Windows Active Directory ^[257]</p> <p>Assign User IDs to Groups ^[260]</p> <p>Single Permissions ^[260]</p> <p>View All ^[261]</p>
2	<p>To identify a new user on this dialog, click on the New button and type in the user's login ID, first name and last name.</p> <p>If required, also provide the user's department name.</p>	
3	<p>To set the user's password, click on the Change Password button.</p> <p>The Change Password dialog displays.</p>	
4	<p>In the New password field, type the user's password.</p> <p>This must be 12 characters or less in length.</p>	
5	<p>In the Retype new field, type the user's password again, for confirmation.</p>	
6	<p>Click on the OK button.</p> <p>A <i>Password Changed</i> message displays.</p>	

Step	Action	See also
7	Click on the OK button.	
8	When you have entered the details for the user, click on the Save button.	
9	Click on either the: <ul style="list-style-type: none"> • New button to add another user, or • Close button to exit the Security Users dialog 	

Notes

- You must have **Security - Manage Users** permission to maintain users, and **Change Password** permission to change the password of the current user; the initial **Admin** administrator automatically has these permissions
- You can transport the user definitions between models as Reference Data, using the **Project | Model Import/Export | Export Reference Data** and **Import Reference Data** options
- If you select the **Accept Windows Authentication** checkbox, when a user opens the model Enterprise Architect checks the user's database for their Windows ID and, if it matches, automatically logs the user in without prompting for a password
- The **Accept Windows Authentication** checkbox enables the **Import** button, which you can select to import user IDs from a Windows Active Directory
- As a security measure, the **Accept Windows Authentication** checkbox is automatically deselected if the project .eap file is moved to a different location; once the file has been relocated, you can select the checkbox again to apply Windows authentication from the new database

Learn more

- [List of Available Permissions](#) ^[264]
- [Export Reference Data](#) ^[308]
- [Import Reference Data](#) ^[308]

3.4.2.4 Import User IDs From Active Directory

You can import the Enterprise Architect security user IDs from Windows Active Directory.

When you import the user IDs, you should create an appropriate user group and assign the user IDs to that group; you can then assign appropriate permissions to the group.

Access **Project | Security | Manage Users**

Use to

- Import Windows user IDs as Enterprise Architect security user IDs for use with Windows Authentication

How to

To import user IDs from Windows Active Directory

Step	Action	See also
1	On the Security Users dialog select the Accept Windows Authentication checkbox and click on the Import button. The Import Users dialog displays.	
2	Click on the down arrow in the Security Group field and select the appropriate security group for the imported user IDs.	
3	Click on the Add button. The Select Users screen displays.	
4	Click on the Object Types button. The Object Types dialog displays.	
5	Select the checkbox for the type of object to import from the Active Directory. Click on the OK button to return to the Select Users dialog.	
6	Click on the Locations button. The Locations dialog displays.	
7	Browse for and select the checkbox for the location to import from within the Active Directory. Click on the OK button to return to the Select Users dialog.	
8	In the Enter the object names to select field, either: <ul style="list-style-type: none"> • Type in the user IDs individually (click on the examples link to see examples of the correct formats) or • Click on the Advance button to search for IDs; the Select Users dialog redisplay showing a Common Queries tab 	
9	In the Name and Description fields, type any characters or text that helps identify the IDs you are searching for.	

Step	Action	See also
10	In the Starts with field, click on the drop-down arrow and select the appropriate qualifier.	
11	If required, select the Disabled accounts or Non-expiring password checkboxes, and/or select a value in the Days since last logon field, to further filter the IDs to search for.	
12	Click on the Find Now button to initiate the search, and to display a list of IDs in the bottom panel of the dialog. You can vary the types of information shown here by clicking on the Columns button and selecting the column headings to display.	
13	When you have identified the IDs to import, click on a required ID (or press (Ctrl) or (Shift) while you click to select several) and click on the OK button. The Select Users dialog redisplay, with the selected ID or IDs listed in the Enter the object names to select field.	
14	Click on the OK button to redisplay the Import Users dialog with the selected users' names listed in the Users panel.	
15	Click on the Import button to add the user IDs to the Security Users dialog.	
16	Click on a user ID to populate the dialog fields with the user ID details, to set group membership and to set single permissions as required.	Group Membership ^[260] Set Single Permissions ^[260]

Notes

- Enterprise Architect generates random passwords for Windows user IDs; however, if required you can assign a new password to an imported user ID

Learn more

- [Maintain Groups](#) ^[262]
- [Change Password](#) ^[270]

3.4.2.5 Assign User To Groups

A user might belong to one or more work groups, each of which would have permissions to perform work in specific areas of Enterprise Architect.

Access [Project | Security | Manage Users: Group Membership](#)

Use to

- Assign a user's ID to one or more work groups

How to

To assign a user to user groups follow the steps below:

Step	Action	See also
1	On the Security Users dialog click on the Group Membership button. The User Groups dialog displays.	
2	Select the checkbox against each group this user belongs to.	
3	Click on the OK button to assign the user to the selected groups.	

Notes

- To assign users to groups, you must have **Security - Manage Users** permission; the initial Admin administrator automatically has this permission.
- User groups must already exist in order to assign users to them

Learn more

- [List of Available Permissions](#) ^[264]
- [Maintain Groups](#) ^[262]

3.4.2.6 Set Up Single Permissions

A user commonly belongs to one or more work groups that give that user group permissions to work in areas of Enterprise Architect. You can also assign additional, specific user permissions to that user ID.

Access [Project | Security | Manage Users: Single Permissions](#)

Use to

- Define the specific permissions an individual user has to perform work in Enterprise Architect

How to

To set up single permissions for a user follow the steps below:

Step	Action	See also
1	On the Security Users dialog, click on the Single Permissions button. The User Permissions dialog displays.	
2	Select the checkbox against each specific permission to assign to this user. Click on the: <ul style="list-style-type: none"> • Select All button to select all permissions for the user, or • Deselect All button to clear all selected permissions 	
3	Click on the OK button to assign the selected permissions to the user.	

Notes

- You must have **Security - Manage Users** permission to assign permissions to users; the initial Admin administrator automatically has this permission
- A user's total permissions are those granted by Group Membership plus those granted by specific permission assignment
- You can transport the user permissions between models as Reference Data, using the **Project | Model Import/Export | Export Reference Data** and **Import Reference Data** options

Learn more

- [List of Available Permissions](#)^[264]
- [Export Reference Data](#)^[308]
- [Import Reference Data](#)^[308]

3.4.2.7 View All User Permissions

This topic explains how to list all the permissions a user has.

Access **Project | Security | Manage Users: View All**

Use to

- Display a list of the permissions a specific user has, derived from their individual profile and from their membership of security groups

How to

Step	Action	See also
1	Select the Manage Users menu option. The Security Users dialog displays.	
2	Select the required user. Click on the View All button. A dialog displays, listing all of the user's permissions.	

3.4.2.8 Maintain Groups

This topic explains how to create a security group. Security groups make it easy to configure sets of permissions and apply them to a number of users in one action.

Access **Project | Security | Manage Groups**

Use to

- Create a security group prior to:
 - Assigning permissions to the group
 - Assigning users to the group

How to

To set up a security group

Step	Action	See also
1	Select the Manage Groups menu option. The Security Groups dialog displays.	
2	In the Group Name and Description fields, type the security group name and a description of the group.	
3	Click on the Save button.	
4	The group name can be selected as the recipient of an internal email; when an email is sent to the group, all members of the group receive that email in their Personal Information window. However, the group name can act as either:	Internal Mail ⁴⁸⁷

Step	Action	See also
	<ul style="list-style-type: none"> • A mail list, in which case each group member receives their own copy of the message, or • A mailbox, in which case the email is a single entity - the group members do not receive separate instances of it; therefore, if one group member responds to or deletes the email, the other group members see that action as if they had performed it themselves <p>To make the group name act as a mailbox, select the Shared Mail checkbox against the group name in the Groups panel.</p> <p>To use the group name as a mail list, leave the checkbox unselected.</p>	

Notes

- You must have **Security - Manage Users** permission to manage user groups; the initial Admin administrator automatically has this permission
- You do not define groups as group logins with passwords; if you intend to use a group login, you can define a single-user login and password that all group members use (that is, Enterprise Architect allows multiple logins under one user ID)
- Emails already sent to a group as a mail list and those sent to a group as a mailbox cannot be interchanged; if you change the status of the **Shared Mail** checkbox, the only way to change the distribution of past emails is to forward them to the group name again
- You can subsequently edit the group name; changes are automatically reflected in the internal Model Mail mail list or mail box

Learn more

- [Maintain Users](#) ^[256]
- [Set Group Permissions](#) ^[263]
- [Assign User To Groups](#) ^[260]
- [List of Available Permissions](#) ^[264]

3.4.2.9 Set Group Permissions

This topic explains how to assign permissions to a security group.

Access **Project | Security | Manage Groups: Set Group Permissions button**

Use to

- Assign permissions to a previously-created user group, to work in various areas of Enterprise Architect

How to

To assign permissions to a security group

Step	Action	See also
1	Select the Manage Groups menu option. The Security Groups dialog displays.	
2	Click on the Set Group Permissions button. The Group Permissions dialog displays.	
3	Select the checkbox against each required permission. Click on the: <ul style="list-style-type: none"> • Select All button to select all permissions • Deselect All button to clear all selected permissions 	
4	Click on the OK button to assign the permissions. All of the users assigned to this group share in this set of permissions.	

Notes

- You can transport the group permissions between models as Reference Data, using the **Project | Model Import/Export | Export Reference Data** and **Import Reference Data** options
- You must have **Security - Manage Users** permission to assign permissions to user groups; the initial **Admin** administrator automatically has this permission

Learn more

- [Maintain Groups](#) ^[262]
- [Assign User To Groups](#) ^[260]
- [Export Reference Data](#) ^[306]
- [Import Reference Data](#) ^[308]
- [List of Available Permissions](#) ^[264]

3.4.2.10 List of Available Permissions

In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled, users can only perform certain tasks if they have the appropriate permissions for those tasks.

Some permissions take precedence over others. For example, if a user has **Use Version Control** permission, they can modify model elements on import even if they do not have **Update Element** permission.

Reference

Permission	Enables the user to	See also
Administer Database	Compact and repair a project database.	Compact a Project [539] Repair a Project [540]
Admin Workflow	Develop and manage workflow scripts.	Workflow scripts [298]
Audit Settings	Change the audit settings in the Audit Settings dialog.	Audit Settings [384]
Audit View	Enable auditing and display data in the Audit View and Audit History tab.	Audit View Controls [389] Audit History Tab [392]
Baselines - Manage	Create, delete, import and export Baselines.	Package Baselines [394]
Baselines - Restore	Merge data into the project model from a Baseline or XML file.	Package Baselines [394]
Change Password	Change your own password or (Administrator) another user's password.	Change Password [270]
Check Data Integrity	Check and repair project data integrity.	Check Project Data Integrity [535]
Configure Datatypes	Add, modify and delete datatypes.	Data Types [987]
Configure Images	Configure alternative element images.	Using The Image Manager [755]
Configure Packages	Configure controlled packages and package properties.	Configure Packages [426]
Configure Resources	Create and manage Resources window items: RTF templates, patterns, profiles and favorites.	Resources [1003]
Configure Stereotypes	Add, modify and delete Stereotypes.	Stereotype Settings [982]
Configure Version Control	Set up version control options for the current model.	Version Control Settings Dialog [346]

Permission	Enables the user to	See also
Export XMI	Export a model to XMI.	Export to XMI ^[413]
Generate Documents	Generate RTF and HTML documents from model packages.	RTF Documents ^[225] HTML Reports ^[235]
Generate Source Code and DDL	Generate source code and DDL from a model element, and synchronize code against model elements if it already exists.	Generate Source Code ^[176] Generate DDL for a Table ^[197] Update Package Contents ^[177]
Import XMI	Import a model from XMI.	Import from XMI ^[417]
Lock Objects	Lock an element or package.	Lock Model Elements ^[27] Lock Packages ^[27]
Manage Diagrams	Create new diagrams, copy and delete existing diagrams, and save a diagram as a UML Pattern.	Create a Pattern ^[127] Add New Diagrams ^[72] Copy a Diagram ^[74] Delete Diagram ^[73]
Manage Glossary	Create, edit and delete glossary items in the Project Glossary.	Project Glossary ^[467]
Manage Issues	Update and delete model Issues.	Project issues ^[462]
Manage Project Calendar	Add, update and delete Project Calendar events; those without this permission can view calendar items.	Calendar ^[505]
Manage Project Information	Update and manage project resources, metrics and risks.	Project Resources ^[449]
Manage Reference Data - Update	Update and delete reference items.	Reference Data ^[98]

Permission	Enables the user to	See also
Manage Replicas	Create and synchronize replicas.	Create Replicas ^[246] Synchronize Replicas ^[247]
Manage Tests	Update and delete Test records.	The Testing Workspace ^[2213]
Reverse Engineer from DDL and Source Code	Reverse engineer from source code or ODBC, and synchronize model elements against code.	Import Source Code ^[1790]
Security - Enable/Disable	Disable user security in Enterprise Architect.	Enable Security ^[253]
Security - Manage Locks	View and delete element locks.	View and Manage Locks ^[268]
Security - Manage Users	Maintain users, groups and assigned permissions.	Maintain Users ^[256] Assign User to Groups ^[260] Set Up Single Permissions ^[260]
Spell Check	Spell check package and set spell check language.	Using the Spell Checker ^[485]
Transfer Data	Transfer model between different repositories.	Perform a Project Data Transfer ^[442]
Transform Package	Perform transformations of packages and elements.	Model Transformations ^[1675]
Update Diagrams	Update diagram appearance, properties and layout, including the Page Setup dialog.	Diagram Properties ^[725] Set Diagram Page Size ^[764]
Update Element	Save model changes (including delete) for elements, packages, and relationships.	Element Tasks ^[795] Packages ^[680] Connector Tasks ^[942]

Permission	Enables the user to	See also
Use Version Control	Check files in and out using version control.	Check In and Check Out Packages ^[353]

3.4.2.11 View and Manage Locks

Enterprise Architect provides a function to view and manage active locks on elements.

Access [Project | Security | Manage Locks](#)

Use to

- Periodically examine and, if necessary, delete locks placed on elements by users

How to

To view locks and, if necessary, delete them

Step	Action	See also
1	Select the Manage Locks menu option. The Active Locks dialog displays.	
2	In the View Locks For panel, click on the radio button for the type of lock to view. <ul style="list-style-type: none"> • All • Groups Only • Users Only Locks of the selected type are listed in the Active Locks panel.	
3	To remove a lock, click on it and click on the Unlock Selected button.	
4	When finished, click on the Close button to close the dialog.	

Notes

- You must have **Security - Manage Locks** permission to view and delete user locks; the initial **Admin** administrator automatically has this permission
- If you want to display the resulting information in a more readable layout, you can resize the dialog and

its columns

- If an element is locked then connectors attached to it are also locked, so to unlock the connector you must unlock the element; however, under certain circumstances you can add new connectors to a locked element

Learn more

- [List of Available Permissions](#)^[264]
- [Add Connectors to Locked Elements](#)^[273]

3.4.2.12 Password Encryption

This topic is retained to support regression to releases of Enterprise Architect prior to version 7.1; for password encryption for all repositories at and beyond release 7.1, you create model shortcuts. Enterprise Architect Security Administrators of projects on SQL Server or Oracle repositories can encrypt the password used to set up the connection between Enterprise Architect and the repository. The Enterprise Architect user does not have the real password, thereby preventing them from accessing the repository using other tools such as Query Analyzer or SQLPlus.

Access **Project | Security | Encrypt Password**

Use to

- Encrypt the password used to set up the connection between Enterprise Architect and a SQL Server or Oracle repository

How to

To encrypt a repository password

Step	Action	See also
1	Select the Encrypt Password menu option. A simple dialog displays.	
2	In the Password field, enter the password to access the repository. The Encrypted field displays the modified password to be used by users.	
3	Instruct users to connect Enterprise Architect to the repository using the encrypted password prefixed with \$\$.	Connect to Oracle Data Repository ^[214] Connect to SQL Server Data Repository ^[212]

Notes

- Do not use the **Test Connection** button as it can cause an error with encrypted passwords

- For SQL Server repositories, you must enter the **Initial Catalog** details from the All tab of the Data Link Properties dialog

Learn more

- [Model Shortcuts](#) ^[183]

3.4.2.13 Change Password

This topic explains how to change a password when security is set.

Access **Project | Security | Change Password** (User)
Project | Security | Manage Users (Administrator)

Use to

- Change your own password (User)
- Set or change any user's password (Administrator)

How to

To change your own user password

Step	Action	See also
1	Select the Change Password menu option. The Change Password dialog displays.	
2	In the Enter old password field, type your current password.	
3	In the New password field, type your new password. This must be 12 characters or less in length.	
4	In the Retype new field, type your new password again, for confirmation.	
5	Click on the OK button. The <i>Password Changed</i> message displays.	
6	Click on the OK button to clear the message. Your new password is effective next time you log in.	

To set or change any user's password as Administrator, follow the steps below

Step	Action	See also
1	Select the Manage Users menu option. The Security Users dialog displays.	
2	In the Users: panel, click on the user's name. The user's details display in the dialog fields.	
3	Click on the Change Password button. The Change Password dialog displays.	
4	In the New password field, type the user's password. This must be 12 characters or less in length. You do not have to enter the user's current password.	
5	In the Retype new field, type the user's password again, for confirmation.	
6	Click on the OK button. The <i>Password Changed</i> message displays.	
7	Click on the OK button.	

Notes

- A user must have **Change Password** permission to change a password; the initial Admin administrator automatically has this permission

Learn more

- [List of Available Permissions](#) ^[264]

3.4.2.14 Lock Model Elements

This topic explains how to set a lock on a model object in **User/Group Locking** mode.

In **Require User Lock to edit** mode, the *Apply a User* lock process applies.

Access

Select:

- **Project Browser | Package context menu | Lock**
- **Project Browser | Diagram context menu | Lock**

- **Project Browser | Element context menu | Lock**
- **Diagram background context menu | Lock** or
- **Diagram Element context menu | Lock**

Use to

- Set an editing lock on a modeling object so that:
 - Nobody can edit the object
 - Only you can edit the object, or
 - Only a member of your user group can edit the object
- Clear any editing lock on the object

How to

To set a lock on an object in **User/Group Locking** mode

Step	Action	See also
1	Select the Lock option on the appropriate context menu. The Element Lock dialog displays.	
2	Select the radio button for the required option: <ul style="list-style-type: none"> • No lock - do not lock this object; clear any existing lock • Full lock - lock this object so that no-one can edit it • User lock - lock this object so that only the locking user can make further edits • Group lock - lock this element so that any member of the specified group can edit the object, but others are excluded 	
3	If you have selected the Group lock option, in the GroupID field click on the drop-down arrow and select the group that can edit the object.	
4	Click on the OK button.	

Notes

- You must have **Lock Objects** permission to lock an element
- If the item already has a lock, only the corresponding lock option and the **No lock** option are shown; you have to release the lock in order to set a different type of lock
- If a diagram is locked and you select an object on it, the object border displays in red, indicating that you cannot change the object

Learn more

- [List of Available Permissions](#)^[264]
- [Security Policy](#)^[255]
- [Apply a User Lock](#)^[274]

3.4.2.15 Add Connectors To Locked Elements

When working in a diagram containing locked elements, you can add connectors depending on the lock status of the source and target elements.

Reference

Element Status	Add Connectors	See also
Source unlocked, target unlocked	Yes, any kind of connector can be added	
Source unlocked, target locked	Yes, except for Composition connectors	
Source locked, target unlocked	No, except for Composition connectors	
Source locked, target locked	No, prohibited for all connectors	

Notes

- Generally, a connector can be added if its source is unlocked - you are modifying what the source can see
- The exception is Composition connectors, where the target (the parent) must be unlocked - you are modifying the parent by adding children
- Connectors with locked source or target elements are also locked; to unlock the connector, you must unlock the source and/or target element

Learn more

- [View and Manage Locks](#)^[268]

3.4.2.16 Lock Packages

This topic explains how to lock the complete contents of a package (and optionally all contents of its child packages) in **User/Group locking** mode, in one operation.

Access **Project Browser | Package context menu | Lock Package**

Use to

- Automatically set or clear locks on elements and diagrams within a package in **User/Group locking** mode, as if the locks had been set or cleared individually

How to

To lock a package

Step	Action	See also
1	Deselect the Project Security Require User Lock to Edit menu option.	Security Policy ^[255]
2	In the Project Browser, right-click on the package to lock. The context menu displays.	
3	Select the Lock Package menu option. The Lock/Unlock Package(s) dialog displays.	
4	In the Lock Type panel, select the appropriate radio button for the lock to apply, as for an individual object lock.	Lock Model Elements ^[271]
5	As required, select the checkboxes to lock: <ul style="list-style-type: none"> Elements and/or diagrams The contents of child packages (that is, lock the whole branch) 	
6	Click on the OK button to apply the lock.	

Notes

- If security is enabled you must have **Lock Objects** permission to lock a package

Learn more

- [List of Available Permissions](#)^[264]

3.4.2.17 Apply a User Lock

In the **Require User Lock to Edit** security mode, where a User Lock is required before any edit can occur, you can set or release the lock in either a diagram or the **Project Browser**.

Access

Select:

- **Project Browser | Package context menu | Apply/Release User Lock**
- **Project Browser | Diagram context menu | Apply/Release User Lock**
- **Project Browser | Element context menu | Apply/Release User Lock**
- **Diagram background context menu | Apply/Release User Lock** or
- **Diagram Element context menu | Apply/Release User Lock**

Use to

- Set an editing lock on a modeling object so that only you can edit the object
- Clear the editing lock on the object

How to

To set or clear a user lock in **Require User Lock to Edit** security mode

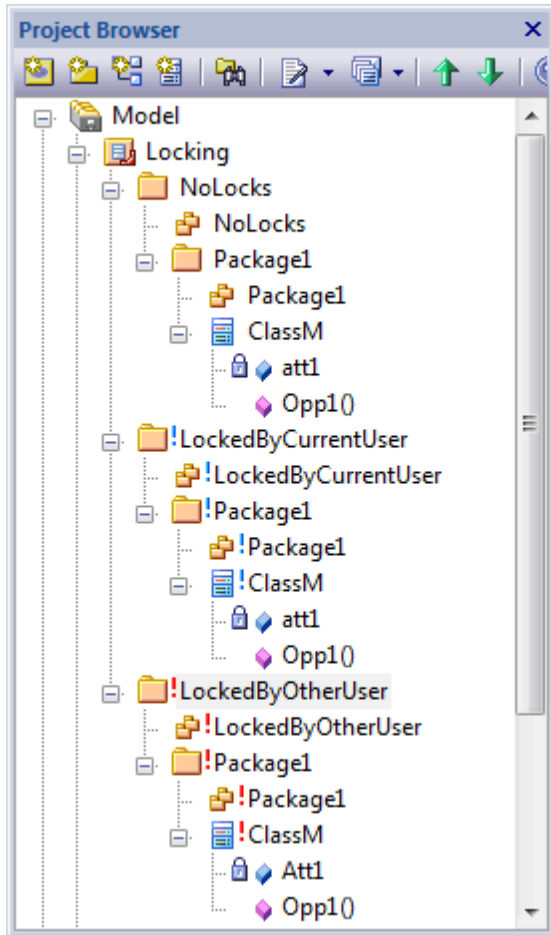
Step	Action	See also
1	Right-click on the required object in the Project Browser or diagram, and select the Apply/Release User Lock option.	
2	Select the appropriate radio button to apply a user lock to, or release a user lock on, the selected object. Enterprise Architect adjusts the lock for the element, or for the diagram and any elements contained in the diagram.	
3	For a package, if required, select to also lock all child packages. If any elements in the package tree are locked by other users, a list of elements that couldn't be locked displays.	

Learn more

- [Security Policy](#)^[255]

3.4.2.18 Locked Element Indicators

When an item is locked through User Security, the lock is indicated in the Project Browser by a marker against the item, as shown below:



Guide:

The meanings of the markers depend on the security mode:

Security Mode	Marker	Meaning	See also
Require User Lock to Edit	No marker	There is no lock, the item is not editable, but any user can now apply a user lock to edit the item.	Security Policy <small>[255]</small> Apply a User Lock <small>[274]</small>
	Blue exclamation mark	The current user has applied a user lock and can edit the item; no other user can edit the item.	
	Red exclamation mark	Another user has applied a user lock, and the current user cannot edit the item. You can find out which user has locked the item.	Identify Who Has Locked An Object <small>[277]</small>

Security Mode	Marker	Meaning	See also
User/Group Lock	No marker	There is no lock, the item is editable, but any user can now apply a user or group lock.	Security Policy [255] Lock Model Elements [277]
	Blue exclamation mark	The item has a lock set by the current user or a group having the current user as a member, and the user can edit the item.	
	Red exclamation mark	The item has a lock set by another user, or a group of which the current user is not a member; the current user cannot edit the item. You can find out which user has locked the item.	Identify Who Has Locked An Object [277]

Notes

- If a diagram is locked and you select an object on it, the object border displays in red; this indicates that you cannot change the object

3.4.2.19 Identify Who Has Locked An Object

If you find that a diagram, package or element is locked, you can find out which group or user currently holds the lock on that item.

Access

- **Project Browser | Package context menu | Lock**
- **Project Browser | Diagram context menu | Lock**
- **Project Browser | Element context menu | Lock**

Use to

- Determine which user or user group has locked the package, diagram or element in which you are interested

How to

To identify the lock holder

Step	Action	See also
1	In the Project Browser, right-click on the diagram, package or element that is locked by another user or user group, and select the Lock option. A message box displays showing which group or user currently holds the lock on	

Step	Action	See also
	that item.	

3.4.2.20 Manage Your Own Locks

You can view and delete your own user-level locks in Enterprise Architect. This is especially useful when working in *Require user locks to edit* security mode.

Access [Project | Security | My Locks](#)

Use to

- Review the locks you have set and, where necessary, clear them

How to

To manage your locks

Step	Action	See also
1	Select the My Locks menu option. The My Locks dialog displays, showing all the locks you have set.	Lock Model Elements ^[271] Apply a User Lock ^[274] Lock Packages ^[273]
2	If you intend to clear locks, either: <ul style="list-style-type: none"> • Select a lock you intend to clear • Press (Ctrl) or (Shift) as you select a number of locks or a range of locks respectively, to clear • Click on Select All to select all of your locks, or • Click on Select None to clear your selection, if you have made an error 	
3	If you have selected locks to clear, click on the Unlock Selected button. The objects that were locked are now unlocked.	

Learn more

- [Security Policy](#)^[255]















3.4.3 Team Review Tools

The Enterprise Architect Project *Team Review* can be used by the team members to discuss the development and progress of a project.

Access [View | Team Review](#) (Ctrl+Alt+U)

Topics

Topic	Detail	See also
Team Review Facility	<p>The Team Review is a discussion forum within your development team community.</p> <p>A point of discussion and the responses to that point are created as <i>Posts</i>. Posts are held in <i>Topic</i> groups, which are in turn organized into <i>Categories</i>; your organization decides on the grouping and organization of Posts.</p> <p>You can create text to explain Categories and Topics, and you can create and respond to Posts and replies</p>	<p>Add a new Category ^[284]</p> <p>Add a New Topic ^[285]</p> <p>Add a New Post ^[287]</p>
Displays	<ul style="list-style-type: none"> The <i>Team Review window</i>, used to create new Categories and Topics and to delete messages, using a context menu <p>Each item in the hierarchy displays a mouse-over tooltip, showing the item title, the author's name and the date and time the item was created</p> <ul style="list-style-type: none"> The <i>Team Review tab</i>, in the main work area, used to view, print, create, edit, and reply to review Posts <p>This tab has a status bar that shows the item author's name, the date and time the item was created, and the date and time the item was last modified</p>	<p>The Team Review Context menu ^[280]</p> <p>Reply to a Post ^[289]</p>
Features	<ul style="list-style-type: none"> Display or edit the text on the Team Review tab by clicking on the item name in the Team Review window Using the Search icon in the Team Review toolbar, search for text strings in the item titles to locate posts on a specific topic Add resources to a Category, Topic or Post; 	<p>Edit an Item ^[291]</p> <p>Search Team Review ^[295]</p> <p>Team Review Resources ^[294]</p> <p>Team Review Options ^[296]</p> <p>Team Review Connections ^[296]</p>

Topic	Detail	See also
	<p>resources such as diagram images and XML files of packages, are held in a Resources folder under the selected Team Review item</p> <ul style="list-style-type: none"> • Associate model elements, diagrams, external files and other postings with a review posting • Change the loading behavior of the forum • Access team reviews from other projects, including those on servers 	Add Object Links ^[292]
Icons	<p>The icons beside the review items have the following meanings:</p> <ul style="list-style-type: none">  Post read  Post unread  Reply read  Reply unread  Status Awaiting Approval (also on Category and Topic icons)  Status Approved (also on Category and Topic icons)  Status Rejected (also on Category and Topic icons)  Category (name in bold indicates category is unread)  Topic read  Topic unread  Resources folder for a Category, Topic or Post  Linked objects folder for a Category, Topic or Post  Diagram or clipboard image within <i>Resources</i> folder  XML File of package, within <i>Resources</i> folder 	<p>Context Menu (Review Status option) ^[283]</p> <p>Add Object Links ^[292]</p>

3.4.3.1 Context Menu

The **Team Review** context menu enables you to access a wide range of functions, depending on the object that you right-click on.

Action	Usage	Shortcut	See also
New Category	Add a new Category, new Topic or new Post to the Team Review.		Add a New Category ^[284]
New Topic	Alternatively, for a Topic or Post, click on the New icon in the window toolbar.		Add a New Topic ^[285]

Action	Usage	Shortcut	See also
New Post			Add a New Post ^[287]
New Category From Template New Topic from Template New Post from Template	Add a new Category, new Topic or new Post, based on a defined template.		Add a New Category ^[284] Add a New Topic ^[285] Add a New Post ^[287]
Post Reply	Create a reply ^[289] to the selected Post. Alternatively, click on the New icon in the window toolbar.		Reply to a Post ^[289]
Rename	Edit the name, in situ, of the currently-selected item.	F2	
Copy Path to Clipboard	Copy the path of the currently-selected item to the clipboard. You can then paste the path into a document or file to add the discussion to any text concerning the item.	Ctrl+C	
Show Contents	Display the description or text of the selected item in the Team Review window, if the Team Review tab is not yet open.		
Share Resource	(If anything other than a Resources folder is selected.) Add a package from the current model, or an image of the currently-displayed diagram, or an image from the clipboard, to the <i>Resources</i> folder under the selected Category, Topic or Post. If the <i>Resources</i> folder does not exist, this option creates it.		Team Review Resources ^[294]
Add Package From Current Model	(If a Resources folder is selected.) Export a package as an XML file from the current model as a resource of the selected Category, Topic or Post. You browse for and select the required package using the Select <item> dialog.		Team Review Resources ^[294] Select <item> Dialog ^[880]

Action	Usage	Shortcut	See also
Import to Current Model	<p>(If a package XML file resource is selected.)</p> <p>Import the resource package to the current model.</p> <p>You browse for and select the required target package using the Select <item> dialog; the resource is imported as a child of that package.</p> <p>This is a useful option for transferring relevant packages from the Team Review of one model into another model.</p>		Select <item> Dialog ^[880]
Add Image of Active Diagram	<p>(If a Resources folder is selected.)</p> <p>Add an image of the currently-displayed diagram as a resource of the selected Category, Topic or Post.</p> <p>You are prompted to provide a reference name for this image.</p>		Team Review Resources ^[294]
Add Active Profiler Report	<p>(If a Resources folder is selected.)</p> <p>Add an active Profiler Report as a resource of the selected Category, Topic or Post.</p> <p>You are prompted to browse for and select the appropriate active report.</p>		Team Review Resources ^[294] Save Report in Team Review ^[2175]
Image From Clipboard	<p>(If a Resources folder is selected.)</p> <p>Add an image held on the clipboard as a resource of the selected Category, Topic or Post.</p>		Team Review Resources ^[294]
View Image	<p>(If a diagram image resource is selected.)</p> <p>Open the View Image window, containing an image of the selected diagram.</p> <p>Alternatively, double-click on the image name.</p>		Team Review Resources ^[294]
Copy Image To Clipboard	<p>(If an image resource is selected.)</p> <p>Copy the image or diagram image to the clipboard.</p>		Team Review Resources ^[294]
Refresh Category 'xyz'	Refresh the currently-open Category, Topic or Post, getting new replies, Posts and Topics that other users might have created.		
Refresh Topic 'xyz'	However, if you open another Category, Topic or Post the Team Review always displays the latest		

Action	Usage	Shortcut	See also
Refresh Post 'xyz'	information from the database. Alternatively, click on the Refresh icon in the window toolbar.		
Reload Current Connection	Reload the entire Team Review connection, getting new Categories, Posts and Topics.		
Review Status	Assign or clear a status marker against the selected Category, Topic or Post; you can mark the item as: <ul style="list-style-type: none"> • Awaiting Approval • Approved • Rejected <p>Or clear the marker (None).</p>		Team Review Tools (Icons) ^[280]
Mark	See the Mark submenu, below.		
Connections...	Access other Team Reviews from other Enterprise Architect models or models located on servers. Alternatively, click on the drop-down arrow in the Connection Options field in the window toolbar, and select one of the listed models. The <Configure Connections> option enables you to add and connect to additional Team Reviews.		Team Review Connections ^[296]
Options...	Change the loading behavior of the Team Review.		Team Review Options ^[296]
Delete Category <xyz> Delete Topic <xyz> Delete Post <xyz> Delete Resource <xyz>	Delete this Category, Topic, Post or reply and all sub-topics and sub-posts, or delete the resource attached to the item.		Edit an Item ^[297]

The Mark Submenu:

Action	Usage	Shortcut	See also
All items as Read	Mark the entire contents of the Team Review as read.		Team Review Tools [279]
All items as Unread	Mark the entire contents of the Team Review as unread.		Team Review Tools [279]
Branch as Read	Mark this item and all its contents as read.		
Branch as Unread	Mark this item and all its contents as unread.		
'xyz' as Unread	Mark only this item as unread.		

3.4.3.2 Add a New Category

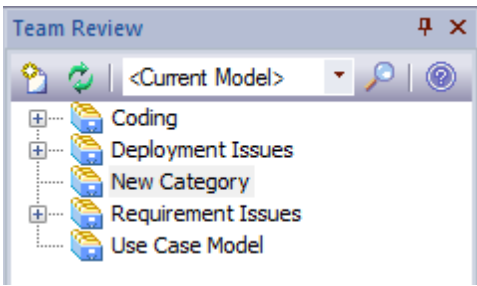
This topic explains how to add new Categories to the Team Review.

Access **View | Team Review (Ctrl+Alt+U)**

Use to

- Create a new Category in which to store related Topics and Posts

How to

Step	Action	See also
1	<p>Right-click on a blank area in the Team Review window and select the New Category context menu option.</p> <p>A new Category icon displays in the hierarchy.</p> 	Context Menu [280]

Step	Action	See also
2	Overtyping the <i>New Category</i> text with the name of the Category, and clicking off the name. The Category description editor displays in the Team Review tab; type the appropriate Category description, if required.	
3	Add new Topics and/or resources to the Category.	Add a New Topic ^[285] Team Review Resources ^[294]
Alternatively		
1	Right-click on a blank area in the Team Review window and select the New Category from template context menu option. A new Category icon displays in the hierarchy.	Context Menu ^[280]
2	Overtyping the <i>New Category</i> text with the name of the Category, and clicking off the name. The Create New Category dialog displays.	
3	Click on the Copy template drop-down arrow and select a predefined template for the Category description. Click on the OK button.	
4	The Category description editor displays in the Team Review tab; type the appropriate Category description, if required.	
5	Add new Topics and/or resources to the Category.	Add a New Topic ^[285] Team Review Resources ^[294]

3.4.3.3 Add a New Topic

This topic explains how to add new Topics to a Category in the Team Review.

Access [View | Team Review](#) (Ctrl+Alt+U)

Use to

- Create a new Topic in which to store related Posts and their responses

How to

Step	Action	See also
1	<p>Either:</p> <ul style="list-style-type: none"> • Right-click on the required Category name in the Team Review window and select the New Topic context menu option • Click on the Category name and click on the New icon in the Team Review toolbar, or • Click on the Category name and press (Ctrl+N) <p>A new Topic icon displays under the Category name.</p>	Context Menu ^[280]
2	<p>Overtyping the <i>New Topic</i> text with the name of the Topic, and clicking off the name.</p> <p>The Topic description editor displays in the Team Review tab; type the appropriate Topic description, if required.</p>	
3	Add new Posts and/or resources to the Topic.	Add a New Post ^[287] Team Review Resources ^[294]
Alternatively		
1	<p>Right-click on the required Category name in the Team Review window and select the New Topic from template context menu option.</p> <p>A new Topic icon displays under the Category Name.</p>	Context Menu ^[280]
2	<p>Overtyping the <i>New Topic</i> text with the name of the Topic, and clicking off the name.</p> <p>The Create New Topic dialog displays.</p>	
3	<p>Click on the Copy template drop-down arrow and select a predefined template for the Topic description.</p> <p>Click on the OK button.</p>	
4	The Topic description editor displays in the Team Review tab; type the appropriate Topic description, if required.	

Step	Action	See also
5	Add new Posts and/or resources to the Topic.	Add a New Post ^[287] Team Review Resources ^[294]

3.4.3.4 Add a New Post

This topic explains how to create a new Post on a Topic in the Team Review.

Access [View | Team Review](#) (**Ctrl+Alt+U**)

Use to



- Create a blank Post
- Create a Post based on a predefined template
- Create a Post from a file link

How to


To create a blank message

Step	Action	See also
1	Either: <ul style="list-style-type: none"> • Right-click on the appropriate Topic name in the Team Review window and select the New Post context menu option • Click on the Topic name and on the New icon in the Team Review toolbar, or • Click on the Topic name and press (Ctrl+N) <p>A new Post icon displays underneath the Topic name.</p>	Context Menu ^[280]
2	Overtyping the <i>New Post</i> text with the name of the Post, and clicking off the name. The Post text editor displays in the Team Review tab.	
3	Type the text of the Post as required.	Edit an Item ^[291]

To create a Post based on a predefined template

Step	Action	See also
1	Right-click on the appropriate Topic name in the Team Review window and select the New Post from template context menu option. A new Post icon displays in the hierarchy.	Context Menu 
2	Overtyping the <i>New Post</i> text with the name of the Post, and clicking off the name. The Create New Post dialog displays.	
3	Click on the drop-down arrow in the Copy template field, and select a predefined template for the Post contents.	
4	Click on the OK button. The template structure is displayed in the Team Review tab.	
5	Enter the text of the Post as required.	Edit an Item 

To create a Post based on a link to an external file

Step	Action	See also
1	Open a file browser (such as Windows Explorer) and locate the file.	
2	Click on the file name and drag it onto the appropriate Topic name in the Team Review window. A new Post icon is created underneath the selected Topic, and the body of the Post is shown in the Team Review tab. A link to the source file is created at the start of the message.	
3	Click twice on the <i>New Post</i> text in the Team Review window, and overtype the text with the name of the Post.	
4	In the Team Review tab, create and edit text around the file link, and add further links if required.	Edit an Item 

When you have created the Post:

Topic	Detail	See also
You can create links from the Post	To elements and diagrams from the: <ul style="list-style-type: none"> • Project Browser • Model Search • Model Views window • Package Browser 	Add Object Links ^[292] The Project Browser ^[567] Model Search ^[611] Model Views ^[598] Package Browser ^[587]
	To related Team Review: <ul style="list-style-type: none"> • Categories • Topics • Posts 	Add Object Links ^[292]
	To external files, either in the text of the Post or on the Category, Topic or Post in the Team Review window (the link is added to the Model Links folder for the Team Review item).	Add Object Links ^[292]
Add resources to the Post		Team Review Resources ^[294]
Other users can reply to it		Reply to a Post ^[289]

3.4.3.5 Reply to a Post

This topic explains how to reply to a Post in the Team Review.

Access [View | Team Review](#) (Ctrl+Alt+U)

Use to

- Create a response to a Post
- Create a response to an existing response to a Post

How to

Step	Action	See also
1	Either: <ul style="list-style-type: none"> • Right-click on the required Post (or response) in the Team Review 	Context Menu ^[280]

Step	Action	See also
	<p>window and select the Post Reply context menu option</p> <ul style="list-style-type: none"> • Click on the Post name and on the New icon in the Team Review toolbar, or • Click on the Post name and press (Ctrl+N) <p>A <i>Re:<Postname></i> entry displays underneath the Post you are replying to, and the cursor becomes active in the Team Review tab to enable you to create and edit your response.</p>	
2	Type in, format and save the contents of the reply.	Edit an Item ^[29]
Alternatively		
1	Open a file browser (such as Windows Explorer) and locate the file.	
2	<p>Click on the file name and drag it into the Team Review window, over the Post to which you are replying.</p> <p>A <i>Re:<Postname></i> entry displays underneath the Post, and the cursor becomes active in the Team Review tab to enable you to edit your response.</p> <p>A link to the source file is created at the start of the message.</p>	
3	Type in, format and save the contents of the reply.	Edit an Item ^[29]

When you have created the Post:

Topic	Detail	See also
You can create links from the Post	<p>To elements and diagrams from the:</p> <ul style="list-style-type: none"> • Project Browser • Model Search • Model Views window • Package Browser 	<p>Add Object Links ^[29]</p> <p>The Project Browser ^[56]</p> <p>Model Search ^[61]</p> <p>Model Views ^[59]</p> <p>Package Browser ^[58]</p>
	<p>To related Team Review:</p> <ul style="list-style-type: none"> • Categories • Topics • Posts 	<p>Add Object Links ^[29]</p>

Topic	Detail	See also
	To external files, either in the text of the Post or on the Category, Topic or Post in the Team Review window (the link is added to the Model Links folder for the Team Review item).	Add Object Links ^[292]
Add resources to the Post		Team Review Resources ^[294]
Other users can reply to it		Reply to a Post ^[289]

3.4.3.6 Edit an Item

This topic explains how to edit or delete a Category, Topic, Post or reply.

How to

Topic	Detail	See also
Select an item to edit	Click on the item name in the Team Review window. The item text displays in the Team Review tab to enable you to edit the text.	
Change the name of the item	Click on the item name in the Team Review window and press (F2) . Overtyping the current name.	
Delete an item	Either: <ul style="list-style-type: none"> Right-click on the item in the Team Review window and select the Delete <item> <name> context menu option, or Click on the item and press (Delete) <p>A confirmation dialog displays; click on the Yes button to remove the item and any dependent items from the Team Review.</p>	Context Menu ^[280]
Edit the text	Access the Team Review Editor standard functions by right-clicking on the text, to display a hierarchy of context menus. When you have completed your editing, right-click on the text	Scroll Through Text ^[2276] File and Print Options ^[2277]

Topic	Detail	See also
	<p>and select the File Save menu option, and then click on another item in the Team Review window to exit the message.</p> <p>The text is saved in the Team Review item.</p> <p>To redisplay the text in the Team Review tab, click on the item name once.</p>	<p>Cut and Paste Options [2279]</p> <p>Image and Object Imports [2282]</p> <p>Character Formatting [2284]</p> <p>Paragraph Formatting [2285]</p> <p>Tab Support [2288]</p> <p>Page Breaks and Repagination [2289]</p> <p>Insert Headers and Footers [2290]</p> <p>Insert Bookmarks [2290]</p> <p>Table Commands [2292]</p> <p>Sections and Columns [2296]</p> <p>Stylesheets and Table of Contents [2297]</p> <p>Text/Picture Frame and Drawing Objects [2302]</p> <p>Search/Replace Commands [2303]</p> <p>Hyperlink From Linked Document [931]</p> <p>Create Elements From Linked Documents [932]</p>

3.4.3.7 Add Object Links

In the Team Review window you can create hyperlinks to elements and diagrams that are associated with a Post.

Use to

- Rapidly navigate to the objects in the Project Browser
- Access the element properties
- Open a diagram directly from the Team Review
- Create links to:
 - Other Categories, Topics and Posts in the Team Review window
 - External files from a file browser

How to

Topic	Detail	See also
Associate an element, diagram, external file or Team Review item with a message	<p>Drag the object over the required Category, Topic of Post in the Team Review window, from the:</p> <ul style="list-style-type: none"> • Project Browser • Package Browser • Model Views window • Model Search dialog or • External file browser <p>To create a link to another Team Review item, press (Ctrl) as you drag that item over the required Category, Topic or Post.</p> <p>If it does not already exist, a <i>Model Links</i> folder is created under the selected Team Review item, and the link to the dragged object is added to the folder.</p>	<p>The Project Browser ^[567]</p> <p>Model Search ^[617]</p> <p>Model Views ^[598]</p> <p>Package Browser ^[587]</p>
Associate an external file with the message text	<p>Click on and drag the file name from any browser into the text of the message.</p> <p>The filename becomes a link to the file; click on it to display the contents of the file.</p> <p>The external file name also becomes a link to the file within the message when you drag the filename onto a Topic to create a Post.</p>	<p>Add a New Post ^[287]</p>

To access each object in the Model Links folder right-click on the object to display the navigation context menu.

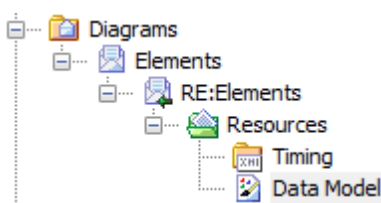
Action	Usage	Shortcut	See also
Open	<p>Open the diagram or external file.</p> <p>Alternatively, double-click on the diagram link.</p>		
Properties	<p>Display the Properties dialog for the selected element or diagram.</p> <p>Alternatively, double click on the element, attribute or operation link to open the Properties dialog of the object.</p>		
Find in all Diagrams	<p>Open the diagram in which the element or diagram is used, or display a list of several diagrams in</p>		

Action	Usage	Shortcut	See also
	which the element or diagram has been used.		
Find in Project Browser	Highlight the element or diagram in the Project Browser.		
	To locate, in the Team Review window, a Team Review item linked to the message, double-click on the link in the Model Links folder.		
Delete Model Link <name>	Delete the association between the message and the object.		

3.4.3.8 Team Review Resources

You can add resources to a Category, Topic or Post within the Team Review window.

Topics

Topic	Detail	See also
Resources	<ul style="list-style-type: none"> • XML files of packages within the current project • Active Profiler reports • Images of currently-active diagrams 	Save and Load Reports ^[217] XML Import and Export ^[41]
Location	<p>Resources are created in a <i>Resources</i> folder underneath the selected Category, Topic or Post, as illustrated below:</p> 	
Management	<p>All Team Review resource management tasks are performed using options on the Team Review context menu.</p> <p>You create the <i>Resources</i> folder by creating a resource underneath the selected Category, Topic or Post.</p> <p>Similarly, you delete the <i>Resources</i> folder by deleting the last resource within it.</p> <p>Having added a resource, you can:</p> <ul style="list-style-type: none"> • Reimport the package XML files to the model or 	Context Menu ^[280]

Topic	Detail	See also
	<ul style="list-style-type: none"> • Display the diagram images 	

3.4.3.9 Search Team Review

The Team Review provides the facility to search the *titles* of all Team Review items.

Use to

- Locate items referring to a specific subject

How to

To search the Team Review

Step	Action	See also
1	In the Team Review window toolbar, click on the Search icon. The search panel displays underneath the toolbar.	
2	In the blank field, type the text string to search for.	
3	If required, select the Match Case checkbox to locate text with the same case as the search string.	
4	If required, select the Match Whole Word checkbox to locate only complete words that match the search string.	
5	Click on the Find button. The search locates the first instance of the search string in the title of a Category, Topic, Post or reply item, and displays the contents of that item in the Team Review tab.	
6	To locate further instances of the text string, click on the Find button again.	
7	To close the search panel, click on the Search icon in the toolbar again.	

3.4.3.10 Team Review Options

You can change the loading behavior of the Team Review, using the Team Review Server Options dialog.

Access **Right-click Team Review window | Options**

Action	Usage	Shortcut	See also
Mark All Unread	Resets all posts you have read to 'unread' (bolds all items in the Team Review).		
Load item data when required	The fastest loading option. Team Review data is only loaded on demand; for example, when you read a post.		
Preload all data	Caches the entire contents of the Team Review on load; this takes longer to load but, once completed, navigating the Team Review is faster.		

3.4.3.11 Team Review Connections

The **Connections** option enables you to access other Team Reviews from other Enterprise Architect models, including models located on servers.

Access **Through the Team Review Toolbar**
From the Team Review context menu options

How to

To switch to another Team Review

Step	Action	See also
1	In the Team Review toolbar, click on the drop-down arrow in the Connection Options field and select the appropriate model name from the list, to connect to the Team Review for that model. If the required model is not listed, click on the <Configure Connections> option; the Team Review Server Connections dialog displays. Go to Step 3 .	
2	<i>Alternatively:</i> Right-click anywhere in the Team Review window and select the Connections context menu option; The Team Review Server Connections dialog displays. In the list in the Connections panel, select the check box against the appropriate model name to connect to the Team Review for that model, and click on the Open button.	

Step	Action	See also
	The connection now switches to the Team Review in the selected model. If the required model is not listed, go to Step 3 .	
3	Select the appropriate Connection Type radio button and click on the New button. <ul style="list-style-type: none"> For a .eap file, a browser dialog displays through which you can search for and select the appropriate model For a model in a DBMS data repository, the Microsoft Data Link dialogs display, which enable you to locate and connect to the repository 	
4	When you have selected and opened or connected to the required Enterprise Architect model, and returned to the Team Review Server Connections dialog, the model name displays in the Connection Name field and in the Connections panel.	
5	Select the check box against the model name and click on the Open button to connect to the Team Review for that model. The Team Review now shows the discussion in the selected model.	

For further details of the fields and buttons on the Team Review Server Connections dialog, refer to the table below.

Action	Usage	Shortcut	See also
Connection Name	Verify the name of the selected model.		
Connection Type	Specify the type of Enterprise Architect model: a local .eap file (as above) or a model on a remote server.		Connect to a Data Repository ^[209]
Target Model	Verify the path to the selected model.		
New	Create a new Team Review connection.		
Delete	Delete the currently selected connection from the Connections list.		
Connections	List all Team Review connections created; click		

Action	Usage	Shortcut	See also
	on the checkbox against the required connection		
Open	Switch the Team Review to the one selected in the Connections list		

3.4.4 Workflow Scripts

Enterprise Architect enables you to create workflow scripts that provide a robust approach to applying company policy and strengthening project development guidelines, by validating against the policy and procedures within the model itself.

Topic	Detail	See also
Project Governance	<p>Good corporate governance relies on well written and transparent project development guidelines and company policy.</p> <p>A project might be compromised if the appropriate policies and procedures are poorly understood and not followed correctly - effective governance can be hampered by human error and the costs of recovering from the inadequate compliance of developers.</p>	
Policies, Procedures and Development	<p>Company policy and procedures can be integrated with the development process to manage work flows, determine access rights, extend role based security permissions and respond to property change events.</p> <p>This approach reduces compliance costs, enhances collaborative development and gives you confidence that projects are being developed correctly the first time around.</p> <p>Development teams can adhere to best practice guidelines that govern model validation, change management, access controls and general development principles.</p>	
Scripts	<p>Project administrators can write scripts to manage the way users interact with a model, such as managing security, staff compliance and model access, and monitoring changes made by users.</p> <p>Administrators can also use workflow scripts to control a user's capacity to change a model element, taking into account factors such as access rights, group membership and even the value of a proposed change.</p>	Workflow Script Functions ^[299]
Script Implementation	<p>When a model is launched, the Workflow Engine is initialized with the current user and group memberships; this information determines who can access and modify parts of a given model.</p> <p>When a selected event occurs, the script engine is initialized with values including the author's name and access rights, and the element name and version details.</p>	

Topic	Detail	See also
	<p>The workflow script implements rules governing change management, access control and model validation; if a user attempts to make changes in violation of company policy, the script denies the update.</p> <p>The user is notified why the validation failed and the activity is logged.</p> <p>These reminders help to reinforce company policy, reduce human error and provide management with valuable project feedback.</p>	

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Admin Workflow** permission to develop and manage workflow scripts

Learn more

- [List of Available Permissions](#) ^[264]

3.4.4.1 Workflow Script Functions

Workflow scripts are executed by the Enterprise Architect workflow engine, to manage user input. You write the scripts in the Scripting window, in VBScript, under the Workflow group type.

Topics

Topic	Detail	See also
Functions for User Input	<p>These are functions that Enterprise Architect calls to validate and control user input.</p> <p>For each of the functions that Enterprise Architect calls, a set of objects are filled.</p>	<p>Functions - Validate and Control User Input ^[300]</p> <p>Filled Workflow Data Structures ^[301]</p>
Functions to create a Search	<p>These are functions that Enterprise Architect calls to create a search with user tasks.</p>	<p>Functions - Create a Search With User Tasks ^[301]</p>
Workflow Data Structures Enterprise Architect fills	<p>These are workflow data structure objects that Enterprise Architect fills.</p>	<p>Filled Workflow Data Structures ^[301]</p>
Workflow Data Structures you fill	<p>These are Workflow data structure objects that you can fill.</p>	<p>Workflow Data Structures You Fill ^[303]</p>

Topic	Detail	See also
Functions you call	These are functions that Enterprise Architect provides for you to call.	Functions You Call [304]

Notes

- If you make changes to a workflow script listed in the Scripting window, click on the **Refresh Scripts** button in the Scripting window toolbar to reload the script with the changes

Learn more

- [Scripting](#) [2372]
- [Script Group Properties](#) [2375]

3.4.4.1.1 Functions - Validate and Control User Input

These are functions that Enterprise Architect calls to validate and control user input. For each function a set of objects is filled.

Function	Action	Return Value
AllowPhaseUpdate(OldValue, NewValue)	Validate a change a user has made to a phase	<ul style="list-style-type: none"> • True to allow this user to make this change • False to disallow the change and revert to the previous value
AllowStatusUpdate(OldValue, NewValue)	Validate a change a user has made to a status	
AllowTagUpdate(TagName, OldValue, NewValue)	Validate a change a user has made to a Tagged Value	
AllowVersionUpdate(OldValue, NewValue)	Validate a change a user has made to a version	
CanEditPhase()	Enable or disable the control for editing a phase	<ul style="list-style-type: none"> • True to allow this user to make changes by enabling the control • False to completely disable edit of this property by disabling the control
CanEditStatus()	Enable or disable the control for editing a status	
CanEditTag(TagName)	Enable or disable the control for editing a Tagged Value	
CanEditVersion()	Enable or disable the control for editing a version	

Function	Action	Return Value
PreAllowPhaseUpdate(OldValue, NewValue)	Determine what information is required to validate this change	Semi-colon separated list of additional data required in order to validate this change <i>Supported Data Type:</i> Tests - fill the Tests array in the WorkflowContext object
PreAllowStatusUpdate(OldValue, NewValue)		
PreAllowTagUpdate(TagName, OldValue, NewValue)		
PreAllowVersionUpdate(OldValue, NewValue)		

Learn more

- [Filled Workflow Data Structures](#) ^[30]

3.4.4.1.2 Functions - Create a Search With User Tasks

These are functions that Enterprise Architect calls to create a search with user tasks.

Function	Action	Return Value
GetWorkflowTasks	Describe the searches that this user must run.	Ignored

3.4.4.1.3 Filled Workflow Data Structures

These workflow data structures are objects that Enterprise Architect fills.

Workflow Data Structure	Description	Calls	See also
WorkflowUser	<p>This object provides information about the user currently logged in to the model.</p> <p>It is filled by Enterprise Architect before any function is called by Enterprise Architect; it has the following properties:</p> <ul style="list-style-type: none"> • Username - the username for login to the system (if using Windows Authentication, this matches the Windows username) • Firstname - as found in the Security 	This object calls the IsMemberOf (GroupName) function (see below).	Maintain Users ^[25]

Workflow Data Structure	Description	Calls	See also
	<p>Users dialog</p> <ul style="list-style-type: none"> • Surname - as found in the Security Users dialog • Fullname - the combination <Firstname > <Surname> (the form Enterprise Architect uses for Author fields and similar) 		
WorkflowContext	<p>This object provides information about the object currently in context.</p> <p>It is filled by Enterprise Architect before any searches except GetWorkflowTasks are run; it has the following properties:</p> <ul style="list-style-type: none"> • MetaType - the type of the current object, either an Enterprise Architect core type or a profile-specified metatype • Name - as found in the object Properties dialog • Status - as found in the object Properties dialog • Phase - as found in the object Properties dialog • Version - as found in the object Properties dialog • Stereotypes - an array of strings for the stereotypes applied to this object • Tags - an array of Tagged Values, providing: <ul style="list-style-type: none"> • Name - the Tagged Value name • Value - the Tagged Value value • Tests - an array of tests; only filled during an <i>Allow*</i> call after the <i>PreAllow*</i> call has specified that tests are required; provides the following details, as found in the Testing window: <ul style="list-style-type: none"> • Name • Status • RunBy 	<p>This object calls the TagValue (TagName) function (see below).</p>	<p>Functions - Create a Search With User Tasks^[30]</p> <p>The Testing Workspace^[2213]</p>

Workflow Data Structure	Description	Calls	See also
	<ul style="list-style-type: none"> • CheckedBy • TestClass • TestType 		

Function	Action	Return Value
IsMemberOf (GroupName)	Check the group membership of the current user.	Returns the value True if the current user is a member of the group with the specified name.
TagValue (TagName)	Get the value from a named tag.	Returns the value of the first Tagged Value with that name, or an empty string if no Tagged Value with that name exists.

3.4.4.1.4 Workflow Data Structures You Fill

These two workflow data structures are objects that you can fill.

Workflow Data Structure	Description
WorkflowStatus	<p>Use this data structure to provide information on the status of the object.</p> <ul style="list-style-type: none"> • LogEntry - set to True or False to indicate whether or not a log item should be recorded • Reason - indicate what reason should be recorded in the log • Action - indicate how to display the log message; valid values are: MessageBox, StatusBar, Output (default)
WorkflowSearches	<p>Use this data structure to provide an array of searches.</p> <p>Use Redim WorkflowSearches(x) to specify the number of searches being provided.</p> <p>Each search has the following attributes:</p> <ul style="list-style-type: none"> • Name - the name of this search • Group - the name of the group that this search should appear under in the Search combo box • ID - the GUID for this search • Tasks - the array of tasks that this search looks for; an entry describes how to find all objects required to meet a particular task:

Workflow Data Structure	Description
	<ul style="list-style-type: none"> • Name - the name of the task, as displayed in the Search view; workflow searches are grouped by this field by default • Conditions - an array of conditions, all of which must be matched for an object to be included in this task; a condition is a comparison of a single field to a value: <ul style="list-style-type: none"> • Column - the name of the field • Operator - operator types, either = (provide matching values only) or <> (provide non-matching values only) • Value - if this contains a comma, the string is treated as a comma separated list of values to compare against; otherwise the string is a single value to compare against

3.4.4.1.5 Functions You Call

These are functions that Enterprise Architect provides for you to call.

Function	Action	Return Value	See also
NewSearch(name, group, guid, taskcount)	Create a new search object to be included in WorkflowSearches . Initialize each member.	The created search	Workflow Data Structures You Fill ^[303]
NewTask(name, conditioncount)	Create a new task object to be included in a search. Initialize each member.	The created task	
NewCondition(column, operator, value)	Create a new condition object to be included in a task. Initialize each member.	The created condition	
SetLastError(message, outputMethod)	Called on user input to the following element properties: <ul style="list-style-type: none"> • Status • Phase • Version, and • Tagged Values It logs and/or reports the provided message to the user. It can be called	The message	

Function	Action	Return Value	See also
	<p>within the functions:</p> <ul style="list-style-type: none"> • AllowPhaseUpdate • AllowStatusUpdate • AllowTagUpdate • AllowVersionUpdate • preAllowPhaseUpdate • preAllowStatusUpdate • preAllowTagUpdate • preAllowVersionUpdate <p>For example:</p> <pre> public function AllowPhaseUpdate (OldValue, NewValue) AllowPhaseUpdate = false Set Last Error " No updating to phase allowed", "messagebox" end function </pre> <p>Parameters: message: Text</p> <p>outputMethod can be "messagebox", "statusbar" or "outputwindow".</p> <p>outputmethod is case sensitive.</p>		

3.4.5 Sharing Reference Data

You can conveniently update your models with reference data (including Glossary and Issue information) by exporting the data to and importing the data from .XML files.

You import data into the model automatically or manually from a reference data .XML file, exported from another model or an iteration of the current model.

Use to (For example)

- Copy glossaries from one model to another
- Add additional stereotype profiles by merging new stereotypes into the model
- Update reference data from files supplied by Sparx Systems as a maintenance release
- Copy resources, clients and so on from one model to another

Learn more

- [Reference Data](#)^[98]
- [Import Reference Data](#)^[308]
- [Export Reference Data](#)^[306]

3.4.5.1 Export Reference Data

This topic explains how to export reference and project data to a custom XML file. This includes table information, filter information, rows and columns.

Access [Project | Model Import/Export | Export Reference Data](#)

Use to

- Capture reference data of one or more specific data types in an XML file, to review or transfer into another project or tool
- The exported data includes all instances of the data type in the project; for example, all defined cardinality values, or all RTF Style Templates

How to

Step	Action	See also
1	Select the Export Reference Data menu option. The Export Reference Data dialog displays.	
2	From the Name list, select the data type tables to export. You can select one or more tables to be exported to a single XML file, by pressing (Ctrl) or (Shift) as you click on the data type table names.	Code and Transform Templates ^[1395] Base Templates ^[1397] Automation Scripts ^[2372] (JavaScript, JScript and VBScript) Cardinality Types ^[986] Constraint Status Types ^[998] CSV ^[435] Specifications Defined Metric Types ^[457] Defined Problem Types ^[1007] Diagram Matrix Profiles ^[753] (Model Profiles) Estimation - Environment Complexity Factor Values ^[518] Estimation - Technical Complexity Factor Values ^[517] General Constraint Types ^[997] General Effort Types ^[456] Import Component Types ^[1894]

Step	Action	See also
		<p>Linked Document Templates ^[934]</p> <p>Macro List ^[1902] (Preprocessor macros)</p> <p>Maintenance Types ^[1001] (Problem Types plus Test Types)</p> <p>Model Authors ^[989]</p> <p>Model Data Types - Code ^[987] and DDL ^[1947]</p> <p>Model Images ^[755]</p> <p>Project Clients ^[993]</p> <p>Project Glossary ^[467]</p> <p>Project Issues ^[462]</p> <p>Project Resources ^[992]</p> <p>Project Roles ^[991]</p> <p>Project Tasks ^[460]</p> <p>Property Types ^[1385] (Tagged Value Types)</p> <p>Requirement Types ^[998]</p> <p>Risk Types ^[458]</p> <p>Scenario Types ^[999]</p> <p>Security - Group Permission ^[263]</p> <p>Security - Groups ^[262]</p> <p>Security User Groups ^[260]</p> <p>Security - User Permissions ^[260]</p> <p>Security - Users ^[256]</p> <p>Standard Complexity Types (a Predefined Reference Data Tagged Value Type ^[1390])</p> <p>Status Colors ^[995] (colors defined for status types)</p> <p>Status Types ^[995]</p> <p>Status Types - Application ^[995] (elements to which status can be applied)</p> <p>Stereotypes ^[1293] (as listed on the Stereotypes page of the UML Types dialog)</p> <p>Templates - HTML Style (the web templates ^[2355] listed in the Resources window)</p> <p>Templates - HTML Style Detail (the content of the HTML report templates)</p>

Step	Action	See also
		<p>Templates - RTF Document (the Extended RTF Style templates^[2272] in the Resources window)</p> <p>Templates - RTF Style (the Legacy RTF Style templates^[2323] in the Resources window)</p> <p>Templates - RTF Style Detail (the content of the Legacy RTF Style templates)</p> <p>Templates - RTF Tag and Language Options (RTF word substitution^[2326] templates)</p> <p>Test Types^[1002]</p> <p>Transformation Templates^[1710]</p>
3	Click on the Export button.	
4	When prompted to do so, enter a valid file name with a .XML extension.	
5	Click on the Save and OK buttons. This exports the data to the file; you can use any text or XML viewer to examine the file.	

Notes

- If there are no instances of a selected data type in the project, the export does not generate any output for that data type in the XML file
- Entries for code language templates and/or transformation templates are included in the **Name** list only if templates for a particular language or transformation exist in the model
- Currently, Standard Complexity Types cannot be directly edited and are therefore effectively standard for all models; they can be listed using the Predefined Reference Data Tagged Value type *ComplexityTypes*

3.4.5.2 Import Reference Data

You can import reference data into a model in Enterprise Architect from a reference data .xml file that was exported from another model or from an iteration of the current model, either:

- Manually, as required, whenever you know there is new or changed data to apply, or
- Automatically whenever the model is reloaded into Enterprise Architect (if the file has changed since the previous import)

When you import data into Enterprise Architect, the system merges the incoming data with the existing data.

If a record already exists it is updated to the new values. If the record does not exist, Enterprise Architect adds a new record. Enterprise Architect never deletes records.

The automatic import checks if the source file has changed since the last import; if the file has not changed, the import does not proceed. If the file has changed, the changed data is imported; however, you can configure Enterprise Architect to display a prompt for you to allow or cancel the import.

Access **Project | Model Import/Export | Import Reference Data > Import File tab**
Project | Model Import/Export | Import Reference Data > Shared File tab



How to

To import reference data *manually*

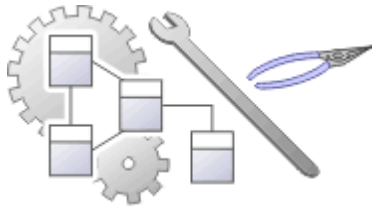
Step	Action	See also
1	Select the Import Reference Data menu option. The Import Reference Data dialog displays.	
2	Click on the Import File tab and on the Select File button, then select the filename to import data from. This must be an XML file produced by the Enterprise Architect Data Exporter.	Export Reference Data ^[306]
3	If you have entered the name of a valid file, a list of available tables to import displays in the Select Datasets to Import panel.	Export Reference Data ^[306]
4	Click on one or more of the tables to import. Press (Ctrl) or (Shift) to click on multiple tables.	
5	Click on the Import button to start the process. A message displays when the import is complete; generally the process is quite fast.	

To import reference data automatically

Step	Action	See also
1	Select the Import Reference Data menu option. The Import Reference Data dialog displays.	
2	Click on the Shared File tab.	

Step	Action	See also
3	If you are changing an existing configuration to import from a different XML file, click on the Clear button to clear the dialog fields.	
4	Click on the Select File button and browse for the filename to import data from. This must be an XML file produced by the Enterprise Architect Data Exporter.	Export Reference Data 
5	If you have entered the name of a valid file, a list of tables to import displays in the Datasets in File panel.	Export Reference Data 
6	If you prefer to control whether or not the automatic import takes place, select the Always prompt before import checkbox.	
7	Click on the Import button to import the reference data now, and to enable the automatic check and import for subsequent reloads.	

3.5 Change Management



This section describes the tools and facilities for controlling and monitoring changes to the data in a project.

Topics

Topic	Detail	See also
Version Control of Packages	<p>Enterprise Architect Model Version Control enables you to:</p> <ul style="list-style-type: none"> • Coordinate sharing of packages between users, with either read-only access or update access, ensuring that work on different areas of the model is coordinated and synchronous rather than conflicting • Save and retrieve a history of changes to packages <p>To use version control in Enterprise Architect, you require a third-party source-code control application such as:</p> <ul style="list-style-type: none"> • Subversion • CVS • MS Team Foundation Server (TFS), or • Any other version control product that complies with the Microsoft Common Source Code Control standard 	Version Control [312]
Tracking Changes	<p>Enterprise Architect provides two separate but complementary facilities for tracking changes to data across the project:</p> <ul style="list-style-type: none"> • Auditing of model changes • Baselining and differencing to capture and roll back changes 	Tracking Changes [38]
Project Data Transfer	<p>Enterprise Architect enables you to transfer project data between project data repositories either for:</p> <ul style="list-style-type: none"> • Sections of the project (XML and CSV) or • The whole project, row by row, table by table (in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect) 	Model Transfer [41]

Topic	Detail	See also

Learn more

- [Check Project Data Integrity](#)^[535]
- [Maintenance](#)^[534]
- [Team Development](#)^[240]
- [Spell Checking](#)^[480]
- [Reference Data](#)^[981]
- [Project Upgrade](#)^[537]

3.5.1 Version Control

Enterprise Architect supports version control of your projects, by allowing you to place your model packages under version control. You can place any individual packages, view nodes or model root nodes under version control.

To use version control in Enterprise Architect, a third-party source-code control application is required that controls access to and stores revisions of the controlled packages.

Guide:

Topic	Detail	See also
Introduction	Version Control provides numerous key facilities: <ul style="list-style-type: none"> • Saving a history of changes to Enterprise Architect packages • The ability to retrieve previous revisions of packages • Propagating model updates between team members • Coordinating sharing of packages between team members 	Introduction ^[313]
Product Setup	To use version control in Enterprise Architect, a third-party source-code control application is required that controls access to and stores revisions of the controlled packages.	Setting Up a Version Control Environment ^[324]
Configuring Enterprise Architect	Once the version control software has been installed and configured, to use your installed version control product you must first define a Version Control Configuration within Enterprise Architect.	Version Control Setup ^[344]
Using Version	Version control activities include such things as;	Using Version Control ^[353]

Topic	Detail	See also
Control	<ul style="list-style-type: none"> Controlling packages Checking out packages for editing Checking in modifications to packages Retrieving previous revisions of controlled packages 	

Notes

- Sparx Systems strongly urge you not to manipulate version controlled package files outside of Enterprise Architect; it is possible to leave the package files in a state that Enterprise Architect cannot recognize
- Database replication should not be combined with version controlled packages
- If the packages under version control contain any alternative images and those images are subject to frequent change, you can set the **Export alternate images** option on the Options dialog to export the images to the version control repository when you check in the packages; if the images are not subject to frequent change, do not select this option and instead use **Export / Import Reference Data** to manage alternative images

3.5.1.1 Introduction

Enterprise Architect's version control integration provides numerous key facilities;

- Saving a history of changes to your model's packages
- The ability to retrieve previous revisions of packages
- Propagating model updates between team members
- Coordinating sharing of packages between team members

This section discusses some factors you should consider when setting up and using version control in your model development.

Topics

Topic	Detail	See also
Version Control Usage	<p>There are two main ways in which Enterprise Architect's projects can be deployed;</p> <ul style="list-style-type: none"> Centralized Shared Model Distributed Private Models <p>Version control is employed in the same way for both scenarios, however, when using Private Model deployment it serves the additional role of propagating model updates throughout the team.</p>	Version Control Usage 315

Topic	Detail	See also
	Version Control can also be used to share standard packages between different projects.	
Version Control Basics	Enterprise Architect must enforce serialized editing of version controlled packages, using the lock-modify-unlock mode of operation.	Version Control Basics ^[318]
Applying Version Control to Models	Using version control in Enterprise Architect consists of placing individual model packages under version control, rather than version controlling the project as a whole.	Version Control Of Model Data ^[316]
Version Control and Project Reference Data	To ensure that changes in reference data are shared between users in a version-controlled project that is deployed as multiple private models, you should periodically export the reference data from the model where the changes were made, and import it into the other models maintained by the team.	Version Control and Reference Data ^[317]
Team Deployment	You should consider the steps involved in setting up a version control environment and applying version control to an Enterprise Architect project to be accessed by a number of users.	Version Control & Team Deployment ^[319]
Project Browser Indicators	Packages under version control are identified in the Project Browser by icons that indicate the current status of the package.	Project Browser Indicators ^[321]
Offline Version Control	<p>You can prevent Enterprise Architect from attempting to make any version control connections by choosing to Work Offline before loading a model.</p> <p>If Enterprise Architect is unable to connect a Version Control Configuration for any reason, it displays warning messages to notify you and provides 'offline' version control functionality for all packages associated with the failed connection.</p>	
System Requirements and Configuration	<p>To use version control in Enterprise Architect, a third-party source-code control application is required that controls access to and stores revisions of the controlled packages.</p> <p>Typically there are:</p> <ul style="list-style-type: none"> • A server component that manages a version control repository, and • Client components on the workstations, that 	System Requirements and Configuration ^[325]

Topic	Detail	See also
	<p>manage local working copies of controlled files</p> <p>Enterprise Architect uses the client component to communicate with the server. A version control client must be installed on every machine where you run Enterprise Architect and want to access your version control system.</p>	

3.5.1.1.1 Version Control Usage

There are four basic ways in which the version control facility might be used:

Field	Usage	See also
Single Shared model	<p>Users share an Enterprise Architect model, stored in a central .EAP file or DBMS repository. This configuration enables you to view changes to other users' packages without explicitly having to check them out, but by simply refreshing your view of the model.</p> <p>Version control is used to;</p> <ul style="list-style-type: none"> • Archive successive versions of your work to date • Maintain package revision history • Provide an "undo" facility to recover from unwanted changes or accidental deletions • Regulate access to packages 	
Multiple Private models	<p>An Enterprise Architect model is created by a single user who configures it for version control. The model file is then distributed to other users, with each user storing their own private copy of the model.</p> <p>Version control is used to;</p> <ul style="list-style-type: none"> • Propagate changes to the model, throughout the team • Archive successive versions of your work to date • Maintain package revision history • Provide an 'undo' facility to recover from unwanted changes or accidental deletions • Regulate access to packages 	
Shared packages	<p>Individual users create separate Enterprise Architect models but share one or more packages.</p> <ul style="list-style-type: none"> • Users share packages through version control 	

Field	Usage	See also
Standard packages	<p>A company might have a standard set of packages which are broadly shared (on a read-only basis).</p> <ul style="list-style-type: none"> Individual users retrieve packages with the Get Package menu option 	

Learn more

- [Version Control Best Practices for Enterprise Architect](#)

3.5.1.1.2 *Version Control of Model Data*

Using version control in Enterprise Architect consists of placing individual model packages under version control, rather than version controlling the project as a whole.

Guide:

Topic	Detail	See also
Version Controlling your Project files	<p>All Enterprise Architect models are stored in databases - even the .EAP file is an MS Jet database.</p> <p>In simple terms, the project file is a single entity of binary data.</p> <p>It is not practical to apply version control to the database as a whole; being binary data, it would require the use of the lock-modify-unlock model of version control, which would mean that only a single user at a time could work on any given (version controlled) model.</p> <p>Therefore, avoid placing your .EAP files under version control, as this can create problems for you.</p> <ul style="list-style-type: none"> Most version control systems mark their controlled files as read only, unless they are specifically checked-out to you The .EAP file is an MS Jet database, and Enterprise Architect must be able to open this file for read/write access when you load your model (Enterprise Architect displays an error message and fails to load the model if it is read-only) 	Version Control Basics ^[318]
Version Controlling Packages in your model	<p>To overcome the limitations described above, Enterprise Architect exports discrete units of the model - the packages - as XML package files, and it is these XML files, not the .EAP file, that are placed under version control.</p> <p>The XML file format used by Enterprise Architect dictates that they too be treated as binary files (therefore it is not possible to merge the XML files either); however, by splitting the model into much smaller parts, this approach enables many users to work on separate parts of the model simultaneously.</p>	

Topic	Detail	See also
Nested version controlled packages	<p>Nested version controlled packages result in much smaller XMI files being exported for parent packages, as the parent packages' XMI files do not contain any content for the version controlled child packages.</p> <p>Version Control of nested packages, together with a model structure having small individual packages, also provides greater scope for multiple users to work concurrently, as individual users are locking much smaller parts of the model.</p>	

3.5.1.1.3 Version Control and Reference data

To ensure that changes in reference data are shared between users in a version-controlled project that is deployed as multiple private models, you should periodically export the reference data from the model where the changes were made, and import it into the other models maintained by the team.

Guide:

Topic	Detail	See also
Reference Data	Reference data is data that is used across a model or project; it is not package-specific.	Reference data ^[304]
Version Control in Enterprise Architect	<p>Version control in Enterprise Architect operates at package level, and therefore does not capture changes in reference data.</p> <p>Where version control is used in a multiple private model set up, changes in reference data are not brought into the model when packages are updated from version control.</p>	
Sharing Reference Data	<p>In a Shared Model environment, all users are accessing the same Project Reference data.</p> <p>However, in a Multiple Private Model environment, to ensure that changes in reference data are shared between users in a version-controlled project, you should periodically export the reference data from the model where the changes were made, and import it into the other models maintained by the team.</p>	Sharing Reference Data ^[305]
Version control of Reference Data	<p>Reference data is exported and imported as an XML file, which contains whatever types of reference data you want to transfer.</p> <p>To place your project reference data under version control, you should export the data as an XML file and apply version control to the file using your version control software, external to Enterprise Architect.</p>	Export Reference Data ^[306]

3.5.1.1.4 Version Control Basics

Enterprise Architect implements version control of your model, by exporting package data from the project database to XML package files. It is these package files that are then placed under version control. The XML file format cannot be merged in the same way that ordinary text files can be merged.

The information below, illustrates why Enterprise Architect must enforce serialized editing of version controlled packages.

Guide:

Topic	Detail	See also
The Lock-Modify-Unlock Solution	<p>Many version control systems use a lock-modify-unlock model to address the problem of different authors in a shared source overwriting each other's work.</p> <p>In this model, the version control repository allows only one person to change a file at a time, and access is managed using locks.</p> <p>Harry must lock a file before he can begin making changes to it. If Harry has locked a file, Sally cannot also lock it, and therefore cannot make any changes to that file. All she can do is read the file, and wait for Harry to finish his changes and release the lock. After Harry unlocks the file, Sally can take her turn in locking and editing the file.</p>	
The Copy-Modify-Merge Solution	<p>Subversion, CVS and a number of other version control systems use a copy-modify-merge model as an alternative to locking.</p> <p>In this model, each user's client contacts the project repository and creates a personal working copy - a local reflection of the repository's files and directories. Users then work simultaneously and independently, modifying their private copies.</p> <p>In due course, the private copies are merged together into a new, final version. The version control system often assists with the merging, but ultimately a person is responsible for making it happen correctly.</p>	
When Locking is Necessary	<p>While the lock-modify-unlock model is generally considered a hindrance to collaboration, there are still times when locking is necessary.</p> <p>The copy-modify-merge model is based on the assumption that files are contextually merge-able: that is, the files in the repository are line-based text files (such as program source code).</p> <p>However, for files with binary formats, such as artwork or sound, it is often impossible to merge conflicting changes. In these situations, it really is necessary for users to take strict turns in changing the file. Without serialized access, somebody ends up wasting time on changes that are ultimately discarded.</p>	

3.5.1.1.5 Applying Version Control in a Team Environment

The following information provides an overview of the steps involved in setting up a version control environment and applying version control to an Enterprise Architect project to be accessed by a number of users.

Use to

- Setup a version control environment
- Apply version control to an Enterprise Architect project

How to

Step	Action	See also
1	Install your version control product.	
2	Create a version control repository.	
3	Create a version control project to be used with your Enterprise Architect project.	
4	<p>Check-out a working copy of the version control project (a module, project or folder within the version control system) into a local folder.</p> <p>(You must do this for every team member that is accessing the version controlled packages, whether you are using a single shared model or each team member stores his own private copy of the model.)</p>	<p>Create a Local Workspace - SVN ^[330]</p> <p>Create a Local Workspace - CVS ^[336]</p>
5	<p>Within Enterprise Architect, define a version control configuration to provide access to the working copy files.</p> <p>Throughout a team, the name of the version control configuration must be the same across all machines. That is, all version control access to a given Enterprise Architect package must be through version control configurations with the same name, across all models and all users.</p> <p>The easiest way to perform this step, (throughout the team), is to have one user set up version control on the model and then share that model with the rest of the team.</p> <ul style="list-style-type: none"> • In Shared Model deployment, all users connect to a single instance of the model database, so the model is shared automatically • In Private Model deployment, it is easiest to distribute copies of the original model (after version control has been set up) to all other members of the team <p>Whenever you open a model (Private or Shared) that uses a version control configuration that is not yet defined on your workstation, Enterprise Architect prompts you to complete the definition for that configuration. This typically means specifying the local working copy directory and maybe choosing the version control project</p>	<p>Version Control Setup ^[344]</p>

	associated with this Enterprise Architect project. Once this has been done, the version controlled packages that already exist in the model are ready for use.	
6	Configure packages within the Enterprise Architect model for version control. That is, apply version control to individual packages.	Configure Package for Version Control ^[362]
7	Check-out and check-in packages as required.	Check Out ^[366] Check In ^[365]

Notes

- It is possible to use multiple version control configurations within the same model, so different packages can still use different version control configurations within the same model, as long as any given package is always accessed via the same version control configuration

Learn more

Team deployment and the use of version control is discussed in two Sparx Systems white papers, available on the Sparx Systems web site:

- http://www.sparxsystems.com/WhitePapers/Version_Control.pdf
- http://www.sparxsystems.com/downloads/whitepapers/EA_Deployment.pdf

3.5.1.1.6 Version Control Nested Packages

In releases of Enterprise Architect later than version 4.5, when you save a package to the version control system only stub information is exported for any nested packages. This ensures that information in a nested package is not inadvertently over-written by a top level package.

Guide:

Topic	Detail	See also
Checking Out	When checking out a package, Enterprise Architect does not modify or delete nested packages; only the top level package is modified. As a consequence of this behavior, if you check out or get a version controlled package with nested packages not already in your model, you see stubs in the model for the nested packages only.	
Get All Latest	If you select the Get All Latest option from the version control menu, Enterprise Architect populates any new stubs from the version control system.	Get All Latest ^[357]

Topic	Detail	See also
Importing Models	<p>You can populate a large and complex model, by 'getting' only the root packages, then using Get All Latest to recursively iterate through the attached and nested packages.</p> <p>This is a powerful and efficient means of managing your project and simplifies handling very large models, even in a distributed environment.</p> <p>The command Import a Model Branch combines the steps described above into a single operation.</p>	Get Package ^[370] Import Controlled Model Branch ^[372]





Notes

- It is recommended you do not mix versions of Enterprise Architect later than version 4.5 with earlier versions when sharing a version controlled model; if this is necessary it is best to go to the Version Control Settings dialog and deselect the **Save nested version controlled packages to stubs only** checkbox, setting Enterprise Architect to the pre-version 4.5 behavior (for the current model only)

3.5.1.1.7 Project Browser Indicators

Packages under version control are identified in the Project Browser by icons that indicate the current status of the package.

Reference

Icon	Indicates that	See also
	This package is version controlled and not checked out to you. You cannot edit the package (unless you check out the package).	
	This package is version controlled and checked out to you. You can edit the package.	
	This package is version controlled, but you checked it out whilst not connected to the version control server. You can edit the package but there could be version conflicts when you check the package in again.	Offline Version Control ^[322]
	This package is controlled and is represented by an XML file on disk, but it is not under version control. You can edit this package.	Controlled Packages ^[422]

Learn more

- [Using Version Control](#) ^[353]

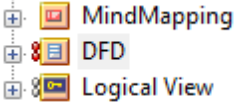
3.5.1.1.8 Offline Version Control

When loading a model that uses version control, Enterprise Architect normally initializes a connection to the version control system for each Version Control Configuration defined in the model.

If Enterprise Architect is unable to connect a Version Control Configuration for any reason, it displays warning messages to notify you and provides 'offline' version control functionality for all packages associated with the failed connection.

Guide:

Topic	Detail	See also
Choose to Work Offline	<p>You can prevent Enterprise Architect from attempting to make any version control connections by selecting the Project Version Control Work Offline menu option before loading a model.</p> <p>This is useful if you know that Enterprise Architect cannot connect to your version control system. For example, if you are working on a laptop computer that is disconnected from your network and you have an Enterprise Architect model that uses a large number of Version Control Configurations, choosing to work offline before you load the model enables you to avoid all the error messages that Enterprise Architect would normally display as each version control connection attempt fails.</p> <p>You can switch between working offline and working online at any time, either before or after a model is loaded. Toggle the Project Version Control Work Offline menu option. Enterprise Architect disconnects or reconnects version control (depending on connection availability) according to your selection.</p>	
Use Version Control Whilst Disconnected From Your Version Control Server	<p>Enterprise Architect 'remembers' the status of a model's version controlled packages. Packages that were checked out to you prior to disconnecting from the server are still shown as checked out to you, even though you are no longer connected to the server. You can still edit these packages as you normally would.</p> <p>Packages that were not checked out to you prior to disconnecting from the server are shown as version controlled and locked. You cannot edit these packages until you check them out.</p>	
Offline Check Out	<p>In releases of Enterprise Architect from release 6.0 onwards, you can 'check-out' and edit a version controlled package even when your machine is disconnected from the version control server. In the example below, the colored 'figure 8' icon for DFD indicates that you have checked it out whilst offline (the gray 'figure 8' icon shown against Logical View indicates that you have checked out a version-controlled package online).</p>	Project Browser Indicators ^[321]

Topic	Detail	See also
	 <p>You should be aware that the version control system - and therefore other users - have no way of knowing that you have 'checked-out' a package whilst offline. It is not possible to merge changes to an XML file that result from two users editing the same package at the same time. If an offline checkout leads to two people editing the same package at the same time, when the changes are brought back online the first-saved set of changes is lost.</p>	
Checking In a Package That Was Checked Out Offline	<p>Once you reconnect your machine to the version control server, if the package you checked out offline is not currently checked out by another user, you can check in that package. However, before Enterprise Architect checks in such a package, it compares the local working copy of the package file with the latest revision in the repository. (These package files remain unchanged in your work area until Enterprise Architect exports the package again before checking in.) If the repository version remains unchanged from when you last updated your local copy, Enterprise Architect exports and checks in your package without further prompting.</p> <p>On the other hand, if the repository now contains a file that has changed since you last updated your local copy, checking in your package overwrites whatever those changes might be. Enterprise Architect displays a message warning you of the pending data loss and giving you the opportunity to abort the check in. At this point, you must decide whether to discard your own changes, using the Undo Check Out command, or continue with your check in and overwrite the changes that have been committed to the repository since you last updated your local copy from the repository.</p> <p>You can use the File Properties command to determine who checked in the last changes to this package. This might help you to discover what changes have been uploaded and decide whose changes take precedence.</p>	
Update Before You Disconnect	<p>Whenever you are connected to the version control server, you are always working with the latest version of a package. This is because you cannot modify a package until you check it out from version control, and checking it out loads the latest revision from the repository into your model.</p> <p>These rules do not apply when you are disconnected from the version control server. You are working on whatever versions you have on your machine, dating back to the last time you updated your local copy of each version controlled package. So, if you are planning to work on a model whilst disconnected from version control, it is a very good idea to make sure that you have the latest versions of all packages</p>	Get All Latest ^[357]

Topic	Detail	See also
	before you disconnect. The Get All Latest option makes this a simple task.	

3.5.1.1.9 Version Control Branching

Currently, Enterprise Architect does not support Version Control Branching.

Work-arounds to achieve similar results might be possible for certain version-control products; contact Sparx Support for advice.

Topic	Link
Registered users	http://www.sparxsystems.com/registered/reg_support.html
Trial users	support@sparxsystems.com

3.5.1.2 Version Control Product Setup

Enterprise Architect relies on third-party version control products to control and maintain the different revisions of your project packages.

Once your version control product is installed and a suitable environment has been created, Enterprise Architect can then utilize that environment to control your project's packages.

Guide:

Topic	Detail	See also
Version Control System Components	<p>Typically, version control products consist of:</p> <ul style="list-style-type: none"> • a server component • a client component <p>Enterprise Architect integrates with version control client components.</p> <p>It provides integration for Subversion, CVS and MS Team Foundation Server command line clients, as well as integration for products having API clients that comply with the MS SCCI specification.</p>	
Version Control Server	<p>The server component maintains the controlled files in their many revisions in a central repository.</p> <p>The server component is usually located on a server machine, that is accessible by all team members who are using version control.</p>	

Topic	Detail	See also
Server Configuration	<p>Steps involved in configuring a version control server usually include:</p> <ul style="list-style-type: none"> • Installing the software • Creating a repository • Creating version control projects (or modules or folders for use with specific projects) • Configuring user IDs and passwords <p>The details of configuring any particular version control server are beyond the scope of this document.</p> <p>Please consult the appropriate documentation.</p>	
Version Control Client	<p>The client component deals with the working copies of the controlled files, submitting or retrieving files to and from the server as required.</p> <p>A version control client must be installed on every machine where you run Enterprise Architect and want to access your version control system.</p>	
Client Configuration	<p>Steps involved in configuring a version control client usually include:</p> <ul style="list-style-type: none"> • Installing the software • Creating a new directory for use as a local working copy folder • Logging in to the version control server • Associating the working copy folder with its corresponding server repository folder <p>For specific information on how to set up a version control environment for use with Enterprise Architect, follow the appropriate link at right.</p>	<p>Create a Subversion Environment^[327]</p> <p>Create a CVS Environment^[335]</p> <p>Create a TFS Environment^[339]</p> <p>Create an SCC Environment^[342]</p>

3.5.1.2.1 System Requirements

Enterprise Architect is a Windows based application and requires a Windows based version control client for integration.

Enterprise Architect is independent of the version control server component and the platform on which that runs.

Reference

Product	Detail	See also
Subversion	<p>Subversion is free, open source software.</p> <p>Subversion server components are available to run on a wide range of different hardware and operating systems; Enterprise Architect is not affected by your choice of server components.</p> <p>Enterprise Architect requires Subversion's Windows-based command line client for integration.</p> <p>There are numerous graphical user interface clients available for use with Subversion, such as TortoiseSVN; this type of client cannot be used for integration with Enterprise Architect, but they can be very useful in preparing a working Subversion environment for use by Enterprise Architect.</p> <p>Binary packages are available for download from:</p> <ul style="list-style-type: none"> • http://subversion.apache.org/packages.html <p>Subversion documentation is available from:</p> <ul style="list-style-type: none"> • http://svnbook.red-bean.com/nightly/en/index.html 	<p>Create a Subversion Environment^[327]</p>
CVS	<p>CVS is free, open source software.</p> <p>CVS server components are available to run on a wide range of different hardware and operating systems; Enterprise Architect is not affected by your choice of server components.</p> <p>Enterprise Architect requires CVS's Windows-based command line client for integration.</p> <p>There are numerous graphical user interface clients available for use with CVS, such as TortoiseCVS; this type of client cannot be used for integration with Enterprise Architect, but they can be very useful in preparing a working CVS environment for use by Enterprise Architect.</p> <p>The software is available for download from:</p> <ul style="list-style-type: none"> • http://www.nongnu.org/cvs/ <p>CVS documentation is available from:</p> <ul style="list-style-type: none"> • http://cvsbook.red-bean.com/cvsbook.html 	<p>Create a CVS Environment^[335]</p>
Microsoft Team Foundation Server	<p>Enterprise Architect is able to use either the command line client for TFS, or the MS TFS-SCC client; your choice of client affects how you specify your Version Control Configuration.</p> <p>MS TFS-SCC clients are available for download from Microsoft's web pages:</p> <ul style="list-style-type: none"> • Visual Studio 2005 Team Foundation Server MSSCCI Provider • Visual Studio Team System 2008 Team 	<p>TFS Settings^[352]</p> <p>SCC Settings^[348]</p>

Product	Detail	See also
	Foundation Server MSSCCI Provider	
SCC compatible products	<p>Any version control product that provides a client that complies with the Microsoft Common Source Code Control standard, version 1.1 or higher, can be integrated with Enterprise Architect.</p> <p>The following products are SCC-compatible and are known to successfully integrate with Enterprise Architect:</p> <ul style="list-style-type: none"> • Accurev • ClearCase • MS Visual Source Safe • MS TFS-SCC • MKS Source Integrity • Perforce • Source Offsite • Snapshot CM <p>Products that do not appear in the list should still integrate successfully with Enterprise Architect, if there is a client available for that product that complies with the MS SCC API specification.</p>	

3.5.1.2.2 Create a Subversion Environment

Before Subversion can be used as a version control provider for Enterprise Architect, the appropriate software must be installed and configured by a Subversion administrator.

The following topics describe some basic tasks that must be completed and some tools for performing these tasks, in order to create an operational Subversion environment.

Topics

Topic	Detail	See also
Install server components	<p>Executable files for Subversion can be obtained from the Apache Software Foundation.</p> <p>Subversion server components are available to run on a wide range of different hardware and operating systems; Enterprise Architect is not affected by your choice of server components.</p> <p>VisualSVN is a package that can greatly simplify the installation, configuration and management of your Subversion server.</p>	Apache Subversion Official Subversion Documentation

Topic	Detail	See also
Create a repository	Please consult the official Subversion documentation.	Official Subversion Documentation
Create Subversion users	Please consult the official Subversion documentation.	Official Subversion Documentation
Create a new repository sub-tree	It is good practice to create folders in your PC's file system for organizing your files and documents; similarly, it is good practice to create a new repository sub-tree in Subversion for each new Enterprise Architect model. TortoiseSVN can greatly simplify the process of creating new repository sub-trees.	Create a Repository Sub-tree ^[329]
Install client components	Executable files for Subversion can be obtained from the Apache Software Foundation.	Apache Subversion
Create a working copy folder	To create a local working copy, you must perform an initial check-out of a folder from the Subversion repository; this downloads a copy of the folder and its contents, to create your local working copy. TortoiseSVN can greatly simplify the initial check out of a working copy folder.	Create a Local Working Copy ^[330]
Subversion under Wine/CrossOver	The set up and use of Subversion with Enterprise Architect under Wine is almost identical to when running natively under Windows, apart from some minor differences when installing the Subversion client and performing the initial check out of the working copy folder.	Subversion under Wine ^[332]
TortoiseSVN	TortoiseSVN is a Windows shell extension; Enterprise Architect can not use TortoiseSVN as its client, it must use the Subversion command line client.	TortoiseSVN ^[334]

Notes

- Enterprise Architect relies on exclusive file locking when applying version control to its packages; file locking was not introduced into Subversion until version 1.2, therefore Enterprise Architect does not work with Subversion releases earlier than Subversion 1.2
- Sparx Systems recommend that each new Enterprise Architect model being added to version control with Subversion should have a separate repository sub-tree created for it, and users should create a new local working copy from the sub-tree to be used with that model

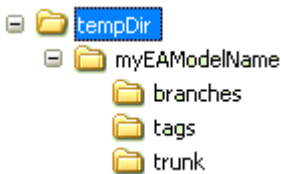
3.5.1.2.2.1 Create a new Repository Sub-tree

In the same way that it is good practice to create folders in your PC's file system for organizing your files and documents, it is good practice to create a new repository sub-tree in Subversion for each new Enterprise Architect model.

Use to

- Create a new sub-tree in the Subversion repository

How to

Step	Action	See also
1	<p>Use Windows Explorer to create a temporary directory structure on your PC file system, that can be imported into the Subversion repository to create a new repository sub-tree. The resulting sub-tree can then be used to control the package files for your Enterprise Architect model.</p> <p>The directory structure should look like this:</p> 	
2	<p>Open a Windows command prompt, navigate to "tempDir" and issue the Subversion command import.</p> <p>For example;</p> <pre>C:\Documents and Settings\user> cd \tempDir C:\tempDir> svn import . https://host.example.com:8443/repos/</pre>	Subversion repository administration basics
3	Delete the temporary directory structure; that is, delete <i>tempDir</i> and all its contents.	

Notes

- After the import is finished, the original tree is not converted into a Subversion working copy; you should delete the temporary structure and check out a fresh working copy of the tree
- The process described above can also be performed using TortoiseSVN's Repository Browser, which provides commands that allow you to simply create new folders directly in the repository

3.5.1.2.2.2 Create a Local Working Copy

In order to make use of Subversion to provide version control of your model's packages, you must prepare a functional SVN working copy folder that can be accessed through an Enterprise Architect version control configuration.

Use to

- Create a Subversion working copy folder, suitable for use with Enterprise Architect

How to

Step	Action	See also
1	Choose a suitable directory on your system, in which to create your Subversion Working Copy.	
2	Open a command prompt window and (if necessary create and then) navigate to a suitable directory to use as your Subversion working copy directory. For example: <pre>C:\> mkdir mySVNWorkSpace C:\> mkdir mySVNWorkSpace/myEAModelName C:\> cd mySVNWorkSpace/myEAModelName</pre>	
3	Issue the command to perform the initial check out from the Subversion repository, specifying the repository folder and local working copy folder, as well as your user name and password. (By specifying your Subversion username and password, you ensure that they are correctly cached by Subversion and available for use by Enterprise Architect). For example; <pre>C:\> svn checkout "https://myserver:8443/svn/repos_folder "C:\mySVNWorkSpace/myEAModelName " --username myUserName --password myPassword</pre> <p>If you specify the HTTPS protocol when performing the initial Subversion check out, a prompt displays to accept a security certificate; in this instance, press (P) to permanently accept the certificate.</p> <p>The nominated local folder is configured as a Subversion working copy folder.</p> <p>Any files already existing in the repository folder are downloaded to the working copy folder as working copy files.</p>	
4	You should now verify that your Subversion environment functions correctly.	Verify the SVN Workspace ^{B337}

Notes

- It is important that Subversion caches your username and password, so that it never has to prompt for user input; a check-out operation might not request authentication, and if it does not you should perform an action that does request authentication, such as adding and committing a dummy test file
- The process described above can also be performed using TortoiseSVN's Checkout command (which provides options to browse) when specifying both your repository folder and your local folder; when prompted for authentication details by TortoiseSVN, make sure you place a check against the **Save Authentication Data** option

3.5.1.2.2.3 Verify the SVN Workspace

After creating the Subversion local working copy to hold the working copies of your Enterprise Architect package files, you should verify that it functions correctly.

You should be able to add files to Subversion, lock the files and commit changes to those files.

Use to

- Verify correct operation of a newly created Subversion working copy folder, before attempting to use it with Enterprise Architect.

How to

Step	Action	See also
1	Use Windows to open a command prompt window.	
2	Change directory to the one you specified as the working copy, in the Subversion checkout command when preparing the SVN workspace. For example; <pre>C:\> cd mySVNWorkSpace</pre>	
3	Create a test file, such as Test.txt, containing the text Subversion Test . For example; <pre>C:\> echo Subversion Test > Test.txt</pre>	
4	Execute the following Subversion commands: <ul style="list-style-type: none"> • <code>svn add Test.txt</code> • <code>svn commit -m"Commit comment" Test.txt</code> • <code>svn update Test.txt</code> • <code>svn lock Test.txt</code> • Use your preferred editor to modify the file and save the changes • <code>svn commit -m"Second commit comment" Test.txt</code> <p>The commands should execute without any errors and without prompting the user</p>	

	for any extra input.	
--	----------------------	--

Notes

- Your environment must be set up such that you can perform these operations without ever being prompted for user ID or password; for further information, please see the *Caching Client Credentials* topic in the official Subversion documentation

3.5.1.2.2.4 Subversion Under Wine-Crossover

The set up and use of Subversion with Enterprise Architect under Wine is almost identical to when running natively under Windows.

When running Enterprise Architect under Wine or CrossOver, you still use a Windows-based Subversion command line client.

There are some differences in the way you go about preparing your Subversion working environment, specifically in the way you install your Subversion client and how you check out a working copy folder from the Subversion repository.

Guide:

Topic	Detail	See also
System Requirements	<p>Sparx Systems has tested and succeeded in using Enterprise Architect with Subversion under Wine 1.2, on Mac OS 10.4 and 10.6.2, as well as Ubuntu 10.04.</p> <p>When using Wine 1.2 on the Ubuntu 9.10 platform, Sparx Systems was unable to successfully use the https: protocol to communicate with the SVN server, however we could use the svn: and file: protocols.</p>	
Installing a Subversion Client	<p>Wine is able to install applications from either a Windows .EXE file, or a .MSI installer file.</p> <p>You should place the installer for your Windows Subversion client in a convenient location on the native file system. Then, open a Wine console window from within Enterprise Architect and run the installer from within the Wine console.</p> <p>This ensures that your Subversion installation accesses the same C: drive and folders that Enterprise Architect is accessing.</p>	Preparing a Subversion Environment Under Wine. ^[332]

Wine is able to install Subversion from either a Windows .EXE file, or a .MSI file. By performing your Subversion installation and initial check out from within a Wine console window that is opened from within Enterprise Architect, you can ensure that you have access to the same C: drive and folders that Enterprise Architect is accessing.

Use to

- Set up Subversion for use with Enterprise Architect, running under Wine

How to

Step	Action	See also
1	Start Enterprise Architect. You do not have to load a project at this stage.	
2	Select Tools Customize > Tools : New . The Customize dialog displays, at the Tools page, and a new, blank entry is created.	Custom Tools ¹⁵⁶
3	Define the new menu item entry, as follows: <ul style="list-style-type: none"> • In the newly-opened Menu contents field, type the name Wine Console • In the Command field, type wineconsole • In the Arguments field, type cmd • The Initial directory field should be left blank 	
4	Click on the Close button. The Customize dialog closes.	
5	Select Tools Wine Console . A Wine console window opens.	
6	Type C: and press (Enter). The Wine console switches to the C: drive.	
7	Issue the command to install your Subversion client. For example; <pre>C:\>/Installers/Subversion-client-1.6.12-1.win32.exe</pre> To install from a .msi file, use Wine's msiexec utility. For example; <pre>C:\>msiexec "Slik-Subversion-1.6.9-win32.msi" /i</pre> Installation of the Subversion command line client begins.	
8	Create a folder to serve as the working copy folder to be used by Enterprise Architect.	

	<p>For example:</p> <pre>C:\>mkdir C:\VC_workspaces\SVN_workcopy</pre>	
9	<p>Issue the command to perform the initial check out from the Subversion repository, specifying the repository folder, the working copy folder, and the username and password.</p> <p>(By specifying your Subversion username and password, you ensure that they are correctly cached by Subversion and available for use by Enterprise Architect).</p> <p>For example;</p> <pre>C:\>svn checkout "https://myServer:8443/svn/repos_folder" "C:\VC_workspaces\SVN_workcopy " "--username myUserName" "--password myPassword"</pre> <p>If the HTTPS protocol is specified when performing the initial Subversion check out, you are prompted to accept a security certificate; in this instance, press (P) to permanently accept the certificate.</p> <p>The nominated local folder is configured as a Subversion working copy folder.</p> <p>Any files already existing in the repository folder are downloaded to the working copy folder as working copy files.</p>	
10	<p>Type Exit and press (Enter).</p> <p>The Wine console window closes.</p> <p>You are now ready to load a project in Enterprise Architect and apply version control to it, following the normal Windows-based procedures.</p>	<p>Version Control Setup^[344]</p> <p>Use Version Control^[353]</p>

Notes

- You should copy the installer for your Windows Subversion client to a convenient location on the native file system, so that you can easily refer to it from within the Wine console window in step 7 above

3.5.1.2.2.5 TortoiseSVN

TortoiseSVN is a Windows shell extension for Subversion. TortoiseSVN provides icon overlays in Windows Explorer that are useful as a tool for observing the status of your Subversion controlled files.

It also enables you to create your repository sub-trees and check out local working copies from within Windows Explorer using simple menu commands.

TortoiseSVN can be downloaded from <http://tortoisesvn.net/downloads.html>.

Notes

- Enterprise Architect must use the Subversion command line client. It **can not use** TortoiseSVN to communicate with the Subversion server

3.5.1.2.3 Create a CVS Environment

Before CVS can be used as a version control provider for Enterprise Architect, the appropriate software must be installed and configured by a CVS administrator.

The following topics describe some basic tasks that must be completed and some tools for performing these tasks, in order to create an operational CVS environment.

Topics

Topic	Detail	See also
Install server components	<p>Executable files for CVS can be obtained from the Free Software Foundation, or from March Hare Software.</p> <p>CVS server components are available to run on a wide range of different hardware and operating systems. Enterprise Architect is not affected by you choice of server components.</p>	FSF CVS March Hare CVS CVS Documentation
Create a repository	Please consult the CVS documentation.	CVS Repository Administration
Create CVS users	Please consult the CVS documentation.	CVS Documentation
Create a new repository module	<p>It is good practice to create folders in your PC's file system for organizing your files and documents. Similarly, it is good practice to create a new repository module in CVS for each new Enterprise Architect model.</p> <p>A repository module represents a project, or set of related files in the repository.</p> <p>TortoiseCVS can greatly simplify the process of creating new repository modules.</p>	CVS Starting a New Project
Install client components	<p>Executable files for CVS can be obtained from the Free Software Foundation, or from March Hare Software.</p> <p>Enterprise Architect is a Windows based application - it requires a Windows based CVS command line client for integration.</p>	FSF CVS March Hare CVS
Create a working copy folder	<p>To create a local working copy, you must perform an initial check-out of a module from the CVS repository. This downloads a copy of the module, to create your local working copy.</p> <p>TortoiseCVS can greatly simplify the initial check out of a working copy folder.</p>	Create a CVS Local Workspace ^[336]
CVS under Wine/ CrossOver	The set up and use of CVS with Enterprise Architect under Wine is almost identical to when running natively under	

Topic	Detail	See also
	Windows, apart from some minor differences when installing the CVS client and performing the initial check out of the working copy folder.	
TortoiseCVS	TortoiseCVS is a Windows shell extension. Enterprise Architect can not use TortoiseCVS as its client, it must use the CVS command line client.	TortoiseCVS ^[338]

Notes

- If you do not already use CVS for version control, you should consider using Subversion instead. Subversion's client-server protocols provide a broader range of possibilities for connecting to remote servers, with easier set up of secure connections
- Sparx Systems recommend that each new Enterprise Architect model being added to version control with CVS should have a separate repository module created for it, and users should create a new local working copy from the module to be used with that model

3.5.1.2.3.1 Prepare a CVS Local Workspace

In order to make use of CVS to provide version control of your model's packages, you must prepare a functional CVS working copy folder, that can be accessed through an Enterprise Architect version control configuration.

Use to

- Prepare a CVS environment, suitable for use with Enterprise Architect

How to

Step	Action	See also
1	Ask your system administrator to install CVS and create a remote repository with a module that you can use to control your Enterprise Architect package files. Your administrator must create a username and password for you before you can make a connection.	
2	Open a command prompt window and (if necessary create and then) navigate to a suitable directory to use as your CVS working copy directory. For example: <pre>C:\> mkdir myCVSWorkSpace C:\> cd myCVSWorkSpace</pre>	
3	Log in to the remote CVS repository.	

	<p>For example;</p> <pre>C:\myCVSWorkSpace> cvs -d:pserver:myUserID@ServerName:/reposPath</pre> <p>Replace <i>myUserID</i> with your CVS username, replace <i>ServerName</i> with the name of your CVS server and replace <i>reposPath</i> with the path to the repository on the server.</p> <p>A prompt for a password displays.</p>	
4	<p>Enter your password.</p> <p>You are logged in to the CVS server.</p>	
5	<p>Perform the initial check out of the CVS repository module, into the local working copy directory.</p> <p>For example;</p> <pre>C:\myCVSWorkSpace> cvs -d:pserver:myUserID@ServerName:/reposPath</pre> <p>(Replace <code>moduleName</code> with the name of the repository module that you want to check out.)</p> <p>A subdirectory is created in your current working directory, that has the same name as the module being checked out.</p> <p>Any files already existing in the repository module are downloaded to the working copy folder as working copy files.</p>	
6	<p>You should now verify that your CVS environment functions correctly.</p>	<p>Verify the CVS Workspace</p> <p>337</p>

Notes

- Much of the process described above can also be performed (more simply) using the TortoiseCVS command **Make New Module**

3.5.1.2.3.2 Verify the CVS Workspace

After creating the CVS local working copy to hold the working copies of your Enterprise Architect package files, you should verify that it functions correctly.

You should be able to add files to CVS, and then commit changes to those files. You should also be able to register as an editor of the file as well as retrieve the list of currently registered editors.

Use to

- Verify correct operation of a newly created CVS working copy folder, before attempting to use it with Enterprise Architect

How to

Step	Action	See also
1	Use Windows to open a command prompt window.	
2	Change directory to the one you specified as the working copy, in the cvs checkout command when preparing the CVS workspace. For example; <pre>C:\> cd myCVSWorkSpace</pre>	
3	Create a test file, such as Test.txt, containing the text CVS Test . For example; <pre>C:\> echo CVS Test > Test.txt</pre>	
4	Execute the following CVS commands: <ul style="list-style-type: none"> • <code>cvs add Test.txt</code> • <code>cvs commit -m"Commit comment" Test.txt</code> • <code>cvs update Test.txt</code> • <code>cvs edit Test.txt</code> • <code>cvs editors Test.txt</code> <p>The commands should execute without any errors and without prompting the user for any extra input.</p> <p>The editors command should produce output that resembles the following;</p> <pre>Test1.txt myUserID Tue Aug 9 10:08:43 2009 GMT myComputer C</pre>	
5	Take note of the userID that follows the filename. Enterprise Architect must find and use this user ID when you create your version control configuration.	CVS Settings 350

Notes

- Your environment must be set up such that you can perform these operations without ever being prompted for input, such as user ID or password

3.5.1.2.3.3 TortoiseCVS

TortoiseCVS is a Windows shell extension for CVS. TortoiseCVS provides icon overlays in Windows Explorer that are useful as a tool for observing the status of your CVS controlled files.

It also enables you to create your repository modules and check out local working copies from within Windows Explorer using simple menu commands.

TortoiseCVS can be downloaded from: <http://www.tortoisecvs.org>.

Notes

- Enterprise Architect cannot use TortoiseCVS to communicate with the CVS server; it must use the CVS command line client

3.5.1.2.4 Create a TFS Environment

Before Enterprise Architect can be used with TFS, the appropriate software must be installed by a TFS administrator. Ask your system administrator to obtain and install the TFS server and client applications.

The following topics describe some basic tasks that must be completed in order to create a TFS environment that can be used by Enterprise Architect

Topics

Topic	Detail	See also
Obtain and install TFS	Enterprise Architect uses the TFS command line client to integrate TFS version control. It is assumed that the TFS command line client is installed as part of your Visual Studio installation.	
Choose a TFS project	It is good practice to create folders in your PC's file system for organizing your files and documents. Similarly, it is good practice to create a new TFS project, or least a new Source Control Folder within a project, for each new Enterprise Architect project. If you have a single Enterprise Architect project, that contains many different models, (for example, a DBMS hosted project with multiple model root nodes), you might choose to create a new TFS project for each separate model. Please consult your TFS documentation.	
Create a TFS workspace	The TFS workspace is used to map a local working folder on your PC, to a Source Control Folder within a TFS project. A single TFS workspace can map many different local folders - each one to a separate Source Control Folder. You can gain performance benefits for some of Enterprise Architect's version control operations if you keep your version controlled package files in a folder that is separate from other artefacts, such as source code files. This can be achieved either by creating a separate work space to use just for your package files, or by creating and mapping a separate folder for package files within an existing workspace.	TFS Workspaces [340]
Configure exclusive check-outs.	The XML file format used for version control of Enterprise Architect's packages can not be merged like ordinary text files. Therefore, Enterprise Architect must enforce serialized editing of its version controlled packages. As a consequence, it is important that TFS is configured to use 'exclusive checkouts' for XML files.	TFS Exclusive Check Outs [342]

Notes

- TFS can also be used with an SCC client. The MS TFS-SCC client is available for download from Microsoft's web site
- Visual Studio Integration (MDG Integration for Visual Studio 2005 or 2008) enhances TFS support by providing access to, for example, work items and bugs within both Enterprise Architect and the MDG Integration product

Learn more

- [System Requirements](#) ^[325]

3.5.1.2.4.1 TFS Workspaces

The TFS workspace is used to map a local working folder on your PC, to a Source Control Folder within a TFS project.

Use to

- Associate a local folder on your PC, with a Source Control Folder in the TFS repository

How to

Step	Action	See also
1	It is assumed a Team Project already exists, that you can use to control your Enterprise Architect package files. The following actions are carried out within MS Visual Studio.	
2	Connect to your TFS server. From the main menu, choose View Team Explorer .	
3	From the main menu, choose View Other Windows Source Control Explorer .	
4	From the main menu, choose Tools Connect to Team Foundation Server . A prompt displays for you to log in to the Team Foundation Server; another prompt then displays to choose an available Team Foundation Server.	
5	Choose an appropriate server and click OK . The selected Team Foundation Server is displayed in the Team Explorer pane. The server's project hierarchy is displayed in the Source Control Explorer's Folders pane.	

6	From the main menu, choose File New Team Project . The New Team Project wizard opens.	
7	Enter an appropriate name and description for the new Team Project. Choose the option to create an empty source control folder.	
8	Create a new Workspace.	
9	In the Source Control Explorer, click on the drop-down arrow in the Workspace field, then choose Workspaces The Manage Workspaces dialog is displayed.	
10	Click on the Add... button. The Add Workspace dialog is displayed.	
11	Enter an appropriate name for the new workspace and enter a comment if required.	
12	Click in the Source Control Folder column, then click the Browse button, to select a Source Control Folder. Select the new folder that was created in step 7.	
13	Click the Browse button in the Local Folder column and create a new local folder. This is the working copy folder into which Enterprise Architect exports the package files.	
14	Click OK . The new workspace is created and saved. The Add Workspace dialog closes.	
15	Click OK . The Manage Workspaces dialog closes.	

Notes

- The local folder referenced in step 13, is the Working Copy Path that should be specified when defining an Enterprise Architect Version Control Configuration to use this TFS workspace

Learn more

- [TFS Settings](#)^[352]

3.5.1.2.4.2 TFS Exclusive Check Outs

The XML file format used for version control of Enterprise Architect's packages can not be merged like ordinary text files. Therefore, Enterprise Architect must enforce serialized editing of its version controlled packages. As a consequence, it is important that TFS is configured to use 'exclusive checkouts' for XML files. Otherwise, TFS can return file statuses that make it look like the package file is not checked-out by another user when indeed it is.

Use to

- Configure TFS to enforce exclusive check outs for XML files

How to

Step	Action	See also
1	Using Visual Studio, from the main menu select View Team Explorer .	
2	In the Team Explorer pane, right-click on the TFS Server name that is controlling the Enterprise Architect package files, then from the context menu select Team Foundation Server Settings Source Control File Types .	
3	Select the entry for XML files (or create an entry if necessary) then click on Edit .	
4	Clear the check-mark from the option Enable file merging and multiple check out .	

3.5.1.2.5 Create an SCC Environment

Before Enterprise Architect can be used with an SCC version control product, the appropriate software must be installed and configured by an administrator.

The following topics describe some basic tasks that must be completed in order to create an SCC based version control environment that can be used by Enterprise Architect.

Topics

Topic	Detail	See also
Install and configure your chosen version control product.	<p>A version control server component is typically installed on a dedicated server machine. All Enterprise Architect users who require access to version control must be able to connect to the server machine.</p> <p>After installing the version control software, the administrator should also create version control user IDs</p>	List of SCC compatible version control providers ^[325]

Topic	Detail	See also
	<p>for all users who require access to version control.</p> <p>Please consult the documentation for your particular version control product.</p>	
Create a new SCC project	<p>It is good practice to create folders in your PC's file system for organizing your files and documents.</p> <p>Similarly, it is good practice to create a new version control project, or least a new folder within a project, for each new Enterprise Architect project.</p> <p>If you have a single Enterprise Architect project, that contains many different models, (for example, a DBMS hosted project with multiple model root nodes), you might choose to create a new version control project for each separate model.</p> <p>Please consult the documentation for your particular version control product.</p>	
Configure your SCC project to support exclusive check-outs for .XML files	<p>The XML file format used for version control of Enterprise Architect's packages can not be merged like ordinary text files. Therefore, Enterprise Architect must enforce serialized editing of its version controlled packages. As a consequence, it is important that your version control system is configured to use 'exclusive checkouts' for XML files.</p>	
Create a local working copy folder	<p>A working copy folder should be created on each users' machine, for Enterprise Architect to use when exporting and importing the version controlled package files.</p> <p>The working copy folder is the 'sandbox' where you modify the controlled files. The working copy folder is usually associated with a folder that exists within the version control repository. Your version control product provides some means by which you associate a working copy folder with a repository folder.</p> <p>It is this folder that is specified as the Local Project Path, when defining your Version Control Configurations.</p> <p>Please consult the documentation for your particular version control product.</p>	Version Control Setup ^[344]

Notes

- When installing the client component software on users' PCs, ensure that the SCC client is also installed, as it might not be a part of the default installation

Learn more

- [Upgrade at Enterprise Architect Version 4.5, Under SCC Version Control](#)^[344]

3.5.1.2.5.1 Upgrade at Enterprise Architect Version 4.5, Under SCC Version Control

When a version-controlled project created under a release of Enterprise Architect earlier than 4.5 is opened in Enterprise Architect release 4.5 or later, you must identify the SCC connection with a new unique ID. You can assign a name to the existing SCC configuration or associate the project with a configuration that has previously been assigned a unique ID.

By having a unique ID for Version Control Configurations, you can assign a configuration quickly and efficiently using configurations that have been created previously for other version controlled repositories. This enables you to configure the many packages to use an existing version control repository; this can apply to packages created for more than just one model enabling a great deal of flexibility.

How to

To upgrade an existing SCC version controlled project created before release 4.5, in Enterprise Architect release 4.5 or later

Step	Action	See also
1	Open the project that has an SCC Version Control Configuration created in Enterprise Architect earlier than version 4.5. The Select or Create Unique ID for Version Control dialog displays.	
2	On the dialog, either create an ID for an existing configuration or choose a previously created one from the Unique ID drop-down list.	
3	The existing SCC configuration is the initial value, represented by SCC-XXXXX ; this number is not especially meaningful, therefore it is recommended that the configuration be given a meaningful name.	
4	You can associate the version controlled package with a previously-defined configuration by selecting an existing configuration from the Unique ID drop-down list (if one exists).	
5	After you have assigned the unique ID, click on the OK button to load the model.	

3.5.1.3 Version Control Setup

Once the version control software has been installed and configured, to use your installed version control product you must first define a Version Control Configuration within Enterprise Architect.

Use to

- Define a new Version Control Configuration, that can then be used to control the packages in your project

How to

Step	Action	See also
1	Start Enterprise Architect and load the model for which you are defining the Version Control Configuration.	
2	Open the Version Control Settings dialog. Project Version Control Version Control Settings	Version Control Settings dialog ³⁴⁶
3	Click on the New button.	
4	In the Unique ID field, type a suitable name.	
5	Select the Type of version control product you are connecting to, by clicking on the corresponding radio button.	
6	At this point, the middle section of the dialog changes to display a collection of fields specific to the type of Version Control Configuration you are defining. Enter details relating to the version control workspace that this configuration is to use.	
7	Click on the Save button. The new configuration is added to the Defined Configurations list.	
8	When you have finished defining your version control configurations, click on the Close button.	

Notes

- Version Control Configuration details are stored in the user's Windows Registry settings, but each project stores a list of the configurations it uses, so that version control connections can be initialized as a project is being loaded
- You can define any number of Version Control Configurations for use in a single model, however any given package can be associated with only one configuration
- If you are using the Corporate or extended editions of Enterprise Architect with security enabled, you must also set up permissions to configure and use version control

Learn more

- [List of Available Permissions](#)^[264]

3.5.1.3.1 Re-use an Existing Configuration

Once a version control configuration has been defined for use in one project, it is possible to re-use that configuration in other projects.

Use to

- Provide access to an already existing version control environment (a working copy directory and its associated repository that is already in use)
- Provide access to version controlled packages that were created (and version controlled) in another project

How to

Step	Action	See also
1	Choose one of; <ul style="list-style-type: none"> • Project Version Control Version Control Settings • Project Browser package control context menu Package Control Version Control Settings The Version Control Settings dialog displays.	Version Control Settings dialog ^[346]
2	Click on the New button. The various fields on the dialog are cleared, ready for data entry.	
3	In the Unique ID field, click on the drop-down arrow and select one of the previously-defined version control configurations. The details of the selected configuration are displayed in the dialog.	
4	Click on the Save button. The configuration details are saved and ready for use in the current project.	

3.5.1.3.2 Version Control Settings Dialog

The Version Control Settings dialog enables you to define Version Control Configurations, which are used by Enterprise Architect to communicate with your version control system.

Access **Project | Version Control | Version Control Settings**

OR **Right-click on any package node | Package Control | Version Control Settings**

Use to

- Define new Version Control Configurations
- Update existing Version Control Configurations

Reference

Field	Usage	See also
This mode is private	<p>Specify whether this model database is to be accessed by just a single user (Private Model) or by multiple concurrent users (Shared Model).</p> <p>The default state is unchecked (Shared Model). If in doubt, use the default value.</p>	
Save nested version controlled packages to stubs only	<p>Specify whether the exported XML file for a version controlled package, shall contain package stubs (place holders) or full content for nested version controlled child packages.</p> <p>It is recommend to always export version controlled child packages as stubs only.</p>	
Report cross package reference deletions before check-in	<p>Choose whether or not Enterprise Architect should alert the user when cross package references are being removed from the exported XML package file, when compared to the previous revision of that file.</p>	Report Deletion of Cross Package References ^[433]
Unique ID	<p>A name that uniquely identifies the configuration.</p> <p>Enter a name to identify a new configuration, or use the drop-down list to retrieve details of a configuration previously defined in a different project.</p>	
Type	<p>Specify the type of version control system associated with this configuration.</p> <p>Once the type is selected, the middle section of the dialog changes to display a collection of fields relating to the type of Version Control Configuration you are defining. For a description of those fields, see the links at right.</p> <p>Set the type to SCC for;</p> <ul style="list-style-type: none"> • MS Visual Source Safe • Rational Clear Case • Perforce • AccuRev • other SCC compatible clients <p>For CVS, Subversion or TFS, choose the type that matches the product you are using.</p>	SCC Settings ^[348] CVS Settings ^[350] SVN Settings ^[351] TFS Settings ^[352]

Field	Usage	See also
New	Create a new Version Control Configuration.	
Save	Save the details of a new or updated configuration.	
Delete	Remove the definitions of the currently selected configuration from this model.	
Defined Configurations	List of configurations that are in use in the current model.	
In future, do not prompt for incomplete configurations	Choose whether or not the user is prompted to complete the definition of configurations that are not fully specified.	
Close	Close this dialog.	
Help	Display this Help topic.	

Notes

- When you first open an Enterprise Architect project that was created by another user and that uses version control, the Version Control Configuration(s) used by that project do not yet exist in your Windows registry settings. You must complete the definitions of those configurations before you can use version control in that project
- It is important that for any given version controlled package file, all access of that file from any model and by any user, is done using Version Control Configurations having the same UniqueID
- If Project Security is enabled, you must have **Configure Version Control** permission to set up version control options for the current model
- It is possible to use multiple version control configurations in the same Enterprise Architect model

Learn more

- [Applying version control to a package.](#)^[362]
- [Applying version control to a hierarchy of packages](#)^[371]
- [Configure Version Control](#)^[264]

3.5.1.3.2.1 SCC Settings

When you specify the type of your Version Control Configuration as SCC, the middle section of the Version Control Settings dialog changes to allow you to enter details that are specific to SCC based configurations.

Access **Project Browser package context menu | Package Control | Version Control Settings: Type:**

SCC

Use to

- Define the working copy folder to be used with an SCC based version control configuration
- Specify the details necessary to connect to the SCC version control system

Reference

Field	Usage	See also
Local Project Path	<p>Displays the full path of the folder that contains the local (working) copies of the XML package files.</p> <p>This field is read-only; its value can only be set through use of the Select Path button.</p>	
Select Path	<p>Choose the Local Project Path, by opening a file browser dialog and navigating through the file system to the appropriate folder.</p> <ul style="list-style-type: none"> • After you choose the appropriate folder path, the Select SCC Provider dialog displays, listing all SCC providers that are installed on the current workstation; choose the SCC provider to use and click on the OK button • At this point, the SCC client opens its own dialog to prompt you for information; SCC products implement this functionality in varied ways, but typically you are prompted to log in to the version control system, then prompted to choose the SCC Project to use and possibly a (server) folder contained within that project • At the conclusion of this process, all of the SCC details should be filled in; you can then save the definition by clicking on the Save button on the Version Control Settings dialog 	<p>System Requirements ^[325]</p> <p>Version Control Settings dialog ^[346]</p>
Current User	<p>Displays the user name used to log on to the version control system that is accessed through this configuration.</p> <p>This field is read-only; the value it displays is retrieved from the SCC client.</p>	
SCC Provider	<p>Displays the name of the SCC provider.</p> <p>This field is read-only; the value it displays is retrieved from the SCC client.</p>	

Field	Usage	See also
SCC Project	Displays the name of the SCC Project that this configuration attaches to. This field is read-only; the value it displays is retrieved from the SCC client.	

3.5.1.3.2.2 CVS Settings

When you specify the type of your Version Control Configuration as CVS, the middle section of the Version Control Settings dialog changes to allow you to enter details that are specific to CVS-based configurations.

Access **Project Browser package context menu | Package Control | Version Control Settings: Type: CVS**

Use to

- Define the working copy folder to be used with a CVS based version control configuration
- Specify the path to the CVS command line client

Reference

Field	Usage	See also
Working Copy Path	Displays the full path of the folder that contains the local (working) copies of the XML package files. This field is read-only; its value can only be set through use of the associated Select Path button.	
Select Path	Choose the Working Copy Path, by opening a file browser dialog and navigating through the file system to the appropriate folder.	
Current User	This field is read-only; its value is retrieved by Enterprise Architect, from a file named <i>CVSRoot</i> , located in the folder specified by Working Copy Path above.	
CVS Exe Path	Displays the full path of the CVS command line client executable file. This field is read-only; its value can only be set through use of the associated Select Path button.	
Select Path	Specify the path to the CVS command line client, by opening a file browser dialog and navigating through the file system to locate the appropriate file.	
VC Time-Out Value	Specify the amount of time that Enterprise Architect waits for a CVS command to complete; if the command does not complete within the	

Field	Usage	See also
	<p>allowed time, Enterprise Architect displays a <i>Time-out</i> error message, detailing the command that failed to complete.</p> <p>This single time-out value is applied to all Version Control Configurations (of types SVN, TFS and CVS) that the user accesses from this workstation.</p>	

Notes

- When connecting to a remote CVS repository, the **Current User** field should display the user name used to log into that repository; if this does not happen, it indicates that Enterprise Architect cannot extract the user name from the file
...*WorkingCopyPath\CVS\Root* and the configuration does not work correctly

3.5.1.3.2.3 SVN Settings

When you specify the type of your Version Control Configuration as Subversion, the middle section of the Version Control Settings dialog changes to allow you to enter details that are specific to Subversion based configurations.

Access [Project Browser package context menu | Package Control | Version Control Settings: Type: Subversion](#)

Use to

- Define the working copy folder to be used with a Subversion based version control configuration
- Specify the path to the Subversion command line client

Reference

Field	Usage	See also
Working Copy Path	<p>Displays the full path of the folder that contains the local (working) copies of the XML package files.</p> <p>This field is read-only; its value can only be set through use of the associated Select Path button.</p>	
Select Path	Choose the Working Copy Path, by opening a file browser dialog and navigating through the file system to the appropriate folder.	
Subversion Exe Path	<p>Displays the full path of the Subversion command line client executable file.</p> <p>This field is read-only; its value can only be set through use of the associated Select Path button.</p>	

Field	Usage	See also
Select Path	Specify the path to the Subversion command line client, by opening a file browser dialog and navigating through the file system to locate the appropriate file.	
VC Time-Out Value	Specify the amount of time that Enterprise Architect should wait for a Subversion command to complete; if the command does not complete within the allowed time, Enterprise Architect displays a <i>Time-out</i> error message, detailing the command that failed to complete. This single time-out value is applied to all Version Control Configurations (of types SVN, TFS and CVS) that the user accesses from this workstation.	

3.5.1.3.2.4 TFS Settings

When you specify the type of your Version Control Configuration as TFS, the middle section of the Version Control Settings dialog changes to allow you to enter details that are specific to TFS based configurations.

Access [Project Browser package context menu | Package Control | Version Control Settings: Type: TFS](#)

Use to

- Define the working copy folder to be used with a TFS based version control configuration
- Specify the user name and password to log in to the TFS server
- Specify the path to the TFS command line client

Reference

Field	Usage	See also
Working Copy Path	Displays the full path of the folder that contains the local (working) copies of the XML package files. This field is read-only; its value can only be set through use of the associated Select Path button.	
Select Path	Choose the Working Copy Path, by opening a file browser dialog and navigating through the file system to the appropriate folder.	
Server Name	Displays the name of the TFS Server that is associated with the working copy folder specified in Working Copy Path above. This field is read-only; Enterprise Architect retrieves the value it displays by querying the TFS client.	

Field	Usage	See also
Workspace Name	Displays the name of the TFS Workspace that is associated with the working copy folder specified in Working Copy Path above. This field is read-only; Enterprise Architect retrieves the value it displays by querying the TFS client.	
User Name	Specify the user name with which to log into the TFS Server.	
Password	Specify the password with which to log into the TFS Server.	
TFS Exe Path	Displays the full path of the TFS command line client executable file. This field is read-only; its value can only be set through use of the associated Select Path button.	
Select Path	Specify the path to the TFS command line client, by opening a file browser dialog and navigating through the file system to locate the appropriate file.	
VC Time-Out Value	Specify the amount of time that Enterprise Architect waits for a TFS command to complete; if the command does not complete within the allowed time, Enterprise Architect displays a <i>Time-out</i> error message, detailing the command that failed to complete. This single time-out value is applied to all Version Control Configurations (of types SVN, TFS and CVS) that the user accesses from this workstation.	

Notes

- If you automatically log in to TFS through means external to Enterprise Architect (for example, through MS Integrated Security), you can leave the **User Name** and **Password** fields blank
- If the **Password** field is blank, Enterprise Architect retrieves the your Windows username and uses that value when determining whether a package file is checked out to you or to another user
- TFS version control can also be accessed using the TFS MSSCCI client; to make use of the TFS MSSCCI client, please define an SCC based Version Control Configuration

3.5.1.4 Use Version Control

Once your version control product is installed and a suitable environment has been created, Enterprise Architect can then make use of that environment to control your project's packages.

The following topics describe the use of Enterprise Architect's version control features.

Guide:

Topic	Detail	See also
Version Control Settings	Version Control Configurations are used by Enterprise Architect to communicate with your version control system. You must define a Version Control Configuration in your project and then use that configuration to control the packages in your project.	Version Control Setup ^[344]
Configure a Package	To put a package under version control, you must mark the package as a controlled package, specify the version control configuration to control it and associate an XML file with the package.	Configure Controlled Package ^[362]
Check In a Model Branch	The Check In a Model Branch command allows you to check-in all packages involved in a particular unit of work, as a single operation.	Check In a Model Branch ^[364]
Check Out a Model Branch	The Check Out a Model Branch command allows you to check-out all packages within a selected model branch, as a single operation.	Check Out a Model Branch ^[365]
Check In a Package	The Check In command allows you to check-in the package currently selected in the Project Browser.	Check In a Package ^[365]
Check Out a Package	To work on a version controlled package you must have the package checked out. The Check Out command allows you to check-out the package currently selected in the Project Browser.	Check Out a Package ^[366]
Undo Check Out of a Package	The Undo Check Out command allows you to undo the check-out of a package, discarding any modifications that have been made by restoring the package content to the latest revision held in version control.	Undo Check Out of a Package ^[367]
Import a Package From Version Control	You can retrieve packages from version control that have been created by other users, or by you in another model, and import them into your current model.	Include Other Users Packages ^[370]
Apply Version Control to a Model Branch	Enterprise Architect enables you to apply version control to all packages within a selected model branch, in a single operation. In this context, a model branch is a package that is currently selected in the Project Browser, and all of the packages contained within it.	Apply Version Control to Branches ^[371]
Export a Version	The Export a Model Branch command exports version	Export Controlled

Topic	Detail	See also
Controlled Model Branch	control information about the root package of a model branch, that is used to simplify the process of exporting and importing a hierarchy of packages from one model to another.	Model Branch ^[371]
Import a Model Branch From Version Control	Model branch files can simplify the process of exporting and importing a hierarchy of packages from one model to another. The Import a Model Branch command uses Enterprise Architect's Model Branch files, of which there are few, to retrieve information about the root package file and import the model branch.	Import Controlled Model Branch ^[372]
View Package Revision History	The File History command allows you to review the change history of version controlled packages. It also enables you to check out a prior revision of the package for editing, effectively allowing you to roll-back to a prior revision of the package.	Review Package History ^[376]
Validate Package Configurations	You can test the validity of the version control settings associated with each version controlled package within your current model.	Validate Package Configurations ^[379]
Resynchronize the Status of Version Controlled Packages	Enterprise Architect enables you to re-synchronize the version control status of version controlled packages within your project with the status reported by your version control provider.	Resynchronize Package Status ^[380]

Notes

- Database replication should not be combined with version controlled packages
- If the packages under version control contain any alternative images and those images are subject to frequent change, you can set the **Export alternate images** option on the Options dialog to export the images to the version control repository when you check in the packages; if the images are not subject to frequent change, do not select this option and instead use Export / Import Reference Data to manage alternative images

Learn more

- [Sharing Reference Data](#) ^[305]

3.5.1.4.1 Project Version Control Menu

Provides access to various project related version control functions

Access **Project | Version Control**

Use to

- Apply version control to a package
- Modify or check version control settings
- Validate settings for all version controlled packages
- Resynchronize the status of all version controlled packages, with their associated version control system.
- Allow working when not connected to version control

Reference

Action	Usage	Shortcut	See also
Configure Current Package	Display the Package Control Options dialog, which enables you to specify whether this package (and its children) is version controlled, and which version control configuration applies.	Ctrl + Alt + P	Package Control Options ^[362] dialog
Version Control Settings	Display the Version Control Settings dialog.		Version Control Settings dialog ^[346]
Validate Package Configurations	Test the validity of the version control settings associated with each version controlled package within your current model.		Validate package configuration ^[379]
Re-Synch Statuses of All Packages	Resynchronize the version control status of packages as recorded in your Enterprise Architect project when they are out of synchronization with the version control status reported by your version control provider. The function acts on all version controlled packages within the Enterprise Architect project, updating the values recorded in the project to match the values reported by the version control provider, without performing any XML import or export.		Resynchronize the version control status of packages ^[380]
Work Offline	Work independently of the version control server. By selecting Work Offline before loading your project, you can prevent Enterprise Architect from attempting to connect to a version control server that is unavailable, thereby avoiding the delays of waiting for the server to respond and any associated error messages.		Offline version control ^[322]

3.5.1.4.2 Package Version Control Menu

The Package Control menu, when invoked from a version controlled package, provides access to all of the version control operations that are performed on packages.

Access Right-click on a version controlled package node | Package Control

Use to

- Open the dialog for working with baselines
- Apply version control to a single package or a selected hierarchy of packages
- Check-in and check-out single packages or a selected hierarchy of packages
- Update packages to the latest revision from the version control repository
- Inspect the revision history or properties of the XML file associated with a package
- Revert a package to a previous revision
- Compare the current model content of a package, against the latest revision of the package in version control
- Import and export hierarchies of packages (model branches) to and from the model, through the version control system
- Synchronize the status of a package, with the version control system

Reference

Action	Usage	Shortcut	See also
Configure	Display the Package Control Options dialog for the currently selected package. Use to apply or remove version control for a package (specify version control settings) or to specify a file for use in XML package control.	Ctrl + Alt + P	Package Control Options Dialog ^[362]
Manage Baselines	Create a Baseline of the current package, or compare the current package with a previous Baseline.	Ctrl + Alt + B	Create a Baseline ^[398]
Check In Branch	Check-in packages contained in the currently selected model branch. For the selected branch of the model, (that is, the selected package and all of its child packages) display the Select Packages to Check In dialog, listing all version controlled packages within that branch that are checked out to you; you can then select packages in the displayed list, to be submitted for check-in.		Check In a Model Branch ^[364]

Action	Usage	Shortcut	See also
	You can also choose to keep the packages checked-out after committing a new revision to version control.		
Check Out Branch	Recursively check out all packages contained within the currently selected model branch, that are version controlled and checked-in (that is, the selected package and all of its child packages).		Check Out a Model Branch ^[365]
Check In	Commit a new revision of the currently selected package to the version control repository and lock the package against further editing. Only available for packages that are checked-out by the current user.		Check In a Package ^[365]
Check Out	Synchronize the currently selected package with the latest revision from the version control repository and unlock the package to allow editing. Only available for packages that are not already checked-out (and whose associated package file is not checked-out).		Check Out a Package ^[366]
Undo Check Out	Restore the selected package to the latest revision in the version control repository and lock the package against further editing.		Undo Check Out of a Package ^[367]
Put Latest	Commit a new revision of the currently selected package to the version control system, while keeping the package checked-out. This is equivalent to checking a package in and immediately checking it back out again. Only available for packages that are checked-out by the current user.		
Get Latest	Synchronize the currently selected package with the latest revision from the version control repository. Available only for packages that are checked in.		Update Selected Package ^[368]

Action	Usage	Shortcut	See also
Get All Latest	<p>Update all of the version controlled packages in the project, to the latest revision retrieved from version control.</p> <p>Only updates packages that are currently checked in.</p> <p>Once the latest revisions are retrieved, Enterprise Architect scans all the controlled packages and fixes any missing cross-references by comparing the package with its XML 1.1 file.</p> <p>If the cross-reference information in the XML does not match the model, Enterprise Architect updates the model with the information from the XML and records this update in the Output window.</p> <p>You can roll back such updates by selecting the entry in the Output Window and using the context menu option Rollback Update (or Rollback Selected Updates if multiple entries are selected).</p> <ul style="list-style-type: none"> • Closing the model clears the entries in the Output window • An entry in the Output window is also cleared as and when you roll-back the update for it 		<p>Update All Packages ^[869]</p> <p>Output Windows ^[164]</p>
Scan XMI and Reconcile Model	<p>Scan the package XMI files associated with each of the project's controlled packages and restore any diagram objects or cross-references that are detected as missing from the project.</p> <p>This function is useful in team environments where each user maintains their own private copy of the model database (i.e. multiple private EAP files) and model updates are propagated through the use of controlled packages; it provides no benefit when the model is hosted in a single shared database that is accessed by all team members.</p> <p>Each controlled package is compared with its associated XMI file and, if the cross-reference information in the model does not match the XMI, Enterprise Architect updates the model with the information from the XMI and records the update in the Output window.</p> <p>You can roll back such updates by right-</p>		<p><i>Version Control Best Practices</i> white paper http://sparxsystems.com/WhitePapers/Version_Control.pdf</p>

Action	Usage	Shortcut	See also
	<p>clicking on the entry in the Output window and selecting the context menu option Rollback Update (or Rollback Selected Updates if multiple entries are selected).</p> <p>Closing the model clears the entries in the Output window; an entry in the Output window is also cleared as and when you roll-back the update for it.</p> <p>This functionality is invoked automatically as part of the Get All Latest operation.</p> <p>When working in an environment that uses a Private Model deployment and your model contains a significant number of cross-package references, it is recommended that you invoke Scan XMI and Reconcile Model from time to time, following the re-importation of controlled packages - for example, after using Get Latest to update a number of packages - or after performing a number of package check-outs.</p> <ul style="list-style-type: none"> • As a general rule, avoid running Scan XMI and Reconcile Model while you have uncommitted changes in your model • Generally, you should: <ul style="list-style-type: none"> • Check-out a number of packages • Invoke Scan XMI and Reconcile Model • Make your modifications • Commit any outstanding changes before you check-out more packages and run Scan XMI and Reconcile Model again 		
File Properties	Display version control properties pertaining to the XMI export file associated with the currently selected package; this also identifies who has checked out the package.		
File History	Display change history information for the currently selected package.		Review Package History ^[376]

Action	Usage	Shortcut	See also
	Revert to or check-out a prior revision of the package.		
Compare with controlled version	Compare the currently selected package with the latest revision of its associated XML file retrieved from version control.		
Add Branch to Version Control	Apply version control to all packages within a selected <i>model branch</i> , in a single operation. In this context, a model branch is a package that is currently selected in the Project Browser, and all of the packages contained within it.		Apply Version Control to Model Branches ^[371]
Export as Model Branch	Export a newly created model branch from your own private copy of a model.		Exporting VC branches ^[371]
Import a Model Branch	Retrieve a model branch and import it into either the source model or another model.		Importing VC branches ^[372]
Get Package	Access packages in the version control repository that are not currently available in your model.		Including Other User's Packages ^[370]
Re-synch Status With VC Provider	Update the version control status value recorded for the selected package in the Enterprise Architect project to match the value reported by the version control provider, without performing an XML import or export. Use this function when the package's version control status recorded in your Enterprise Architect project is out of synchrony with the version control status reported by your version control provider.		Match the value reported by the version control provider ^[380]
Version Control Settings	Display the Version Control Settings dialog.		Version Control Settings dialog ^[346]
Update Package Status	Provide a bulk update on the status of a package, including status options such as Proposed , Validate and Mandatory . This command is not specific to version controlled packages.		

Action	Usage	Shortcut	See also

Notes

- If the selected package is not under version control, this menu displays a number of different options

Learn more

- [Controlled Package Menu](#)^[424]

3.5.1.4.3 Configure Controlled Package

To put a package under version control, you must mark the package as a controlled package, specify the version control configuration to control it and associate an XMI file with the package. Enterprise Architect can then export and import the package data to and from the file as required and issue commands to the version control system as appropriate.

Use to

- Apply version control to a single package

How to

Step	Action	See also
1	In the Project Browser, right-click on the package to be placed under version control and choose Package Control Configure... The Package Control Options dialog displays.	
2	Click on the Control Package checkbox to insert a check mark, indicating that this package is to be controlled.	
3	Click on the Version Control drop-down arrow and select the Version Control Configuration to be used to control this packages. The XMI Filename field then displays a default filename for the package export file, based on the package name.	
4	Optionally, modify the default XMI Filename . Choose either; <ul style="list-style-type: none"> • type a new name into the filename field • click on the browse button (...) to open a file chooser dialog. <p>The target file must be located within the working copy folder of the selected Version Control Configuration, or one of its sub-folders.</p>	

Step	Action	See also
5	<p>Optionally, update the Version ID for this package.</p> <p>The Version ID field defaults to 1.0; if necessary, change this to the appropriate reference.</p>	
6	<p>Optionally, update the Owner field.</p> <p>The Owner field defaults to your user name; if necessary, type or select the name of the user who owns the package.</p>	
7	<p>Click on the OK button.</p> <p>The Add Package to Version Control dialog displays.</p>	
8	<p>Optionally, clear the Keep checked out checkbox.</p> <p>After applying version control, the package is either checked-out for editing, or checked-in and locked against editing, in accordance with this setting.</p>	
9	<p>Click on the OK button.</p> <p>The Add Comment dialog displays.</p>	
10	<p>Optionally, add any further comments to the default comment.</p> <p>Enterprise Architect provides a default comment that includes the current date & time.</p>	
11	<p>Click on the OK button.</p> <p>The current package is exported to the nominated XMI file, which is then committed to version control. The package icon in the Project Browser is updated to reflect the package's version control status.</p>	Project Browser Indicators ³²⁴

Notes

- If you are using the Corporate or extended editions of Enterprise Architect with security enabled, then this feature is only available to users who have been granted permission to configure and use version control

Learn more

- [List of Available Permissions](#) ²⁶⁴

3.5.1.4.4 Check In a Model Branch

The Check In a Model Branch command enables you to check-in all packages involved in a particular unit of work, as a single operation. You can even commit new revisions of the affected packages as you complete milestones, while still keeping the packages checked-out for further editing.

Use to

- Check in a selection of packages from within the currently selected model branch

How to

Step	Action	See also
1	<p>Right-click on the package icon at the root of the model branch that is to be checked in and select Package Control Check In Branch.</p> <p>The Select Packages to Check-in dialog displays, listing all version controlled and checked-out packages within that model branch.</p>	<p>Check In Package^[365]</p> <p>Check Out Model Branch^[365]</p>
2	<p>Select the packages to be checked-in. (By default, the entire list is selected.)</p> <p>Optionally;</p> <ul style="list-style-type: none"> • Click an individual package to select just that package • Ctrl + click to add or remove an individual package from the selection • Shift + click to select a range of packages • Click on All to select all listed packages • Click on None to clear the selection 	
3	<p>Optionally, tick the checkbox Keep packages checked-out after committing new revision.</p> <p>Use this option to commit into version control, a new revision of all selected packages, while keeping all packages checked out for further editing.</p>	
4	<p>Click on the OK button.</p> <p>The Add Comment dialog displays. A default comment is provided that contains the current date and time.</p>	
5	<p>Optionally, modify the default check-in comment.</p> <p>This comment is applied to all packages that are checked in.</p>	
6	<p>Click on the OK button.</p> <p>The selected packages are exported and checked-in. The package icons</p>	

Step	Action	See also
	are updated to reflect any change in version control status. If you opted to keep packages checked out, there is no change in status.	

3.5.1.4.5 Check Out a Model Branch

The Check Out a Model Branch command allows you to check-out all packages within a selected model branch, as a single operation.

Access [Project Browser package context menu | Package Control | Check Out Branch](#)

Use to

- Check out an entire sub-tree of model packages

How to

Step	Action	See also
1	Right-click on the package icon at the root of the model branch that is to be checked out and select the Check Out Branch context menu option. The selected package and all of its contained sub-packages are recursively checked out. Any packages that cannot be checked-out are listed in a message box, with a brief description of the problem. <i>The package is already checked out by user 'Fred'</i> .	Check Out Package ^[366] Check in Model Branch ^[364]
2	When Project Security is enabled in <i>Lock to Edit</i> mode, Enterprise Architect prompts you to apply a User Lock throughout the selected model branch before proceeding.	

Notes

- Any packages that cannot be checked-out are listed in a message box, with a brief description of the problem. For example: The package is already checked out by user 'Fred'

3.5.1.4.6 Check In a Package

The Check In command allows you to check-in the package currently selected in the Project Browser.

Use to

- Check in a single package

How to

Step	Action	See also

1	<p>In the Project Browser, right-click on the package that is to be checked in and select Package Control Check In.</p> <p>The package is exported and the Add Comment dialog displays. A default comment is provided that contains the current date and time.</p>	<p>Check In Model Branch ^[364]</p> <p>Check Out Package ^[365]</p>
2	Optionally, modify the default check-in comment.	
3	<p>Click OK.</p> <p>The package file is checked-in to version control and the package icon is updated to reflect the change in version control status.</p>	

3.5.1.4.7 Check Out a Package

To work on a version controlled package you must have the package checked out. When a package is checked out to a specific user, no other user can check out the package to make changes to it until it has been checked in again.

The Check Out command allows you to check-out the package currently selected in the Project Browser.

Use to

- Check out a single package

How to

Step	Action	See also
1	<p>In the Project Browser, right-click on the package that is to be checked out and select Package Control Check Out.</p> <p>The local XML file associated with the package is checked-out from version control.</p>	<p>Check Out Model Branch ^[365]</p> <p>Check In Package ^[365]</p>
2	<p>When working in a Private Model, if Enterprise Architect determines that the package content in the model is already up to date with the latest revision of the package file retrieved from version control, then the Import Package dialog displays. This dialog is not displayed for Shared models.</p> <p>The following options are offered;</p> <ul style="list-style-type: none"> • Force reload From XML • Accept current package <p>Choose Accept current package to skip the process of re-importing the package from XML.</p> <p>The option Refresh model view allows you to specify whether or not to refresh the Project Browser and diagrams, by reloading the package content from the project database.</p> <p>The Import Package dialog also has an option Always use these settings. When selected, if you subsequently check out a package that is found to</p>	

	<p>be up to date, the same settings are applied again without displaying the dialog.</p> <ul style="list-style-type: none"> To display the dialog if, for example, you want to change the settings, press a Ctrl key while you select the Package Control Check Out menu option. 	
3	<p>The package file is imported into your model.</p> <p>The package icon is updated to reflect the change in the package's version control status.</p>	

Notes

- If you check out a version controlled package whilst offline, the package icon has a red figure **8** in front of it

Learn more

- [Offline Version Control](#)^[322]

3.5.1.4.8 Undo Check Out of a Package

The Undo Check Out command allows you to undo the check-out of a package, discarding any modifications that have been made by restoring the package content to the latest revision held in version control. The package returns to a Checked-in state and subsequently can be checked out by any user.

Use to

- Undo the Check out of the selected package, restoring the package content to the latest revision held in version control

How to

Step	Action	See also
1	<p>In the Project Browser, right-click on the package and select Package Control Undo Check Out.</p> <p>A confirmation dialog is displayed.</p>	
2	<p>Click OK.</p> <p>The latest revision of the package is retrieved from version control and re-imported into your model.</p> <p>The package icon is updated to reflect the change in the package's version control status.</p>	

3.5.1.4.9 Update to the Latest Revision of Selected Package

When you are part of a team working in a Distributed Model environment, from time to time, you will want to update your model with the changes that other team members have committed into version control. The Get Latest command allows you to bring in updates made by other users, for the package currently selected in the Project Browser.

Use to

- Update a single file package to the latest revision retrieved from version control

How to

Step	Action	See also
1	<p>In the Project Browser, right-click on the package that is to be updated and select Package Control Get Latest.</p> <p>The local XMI file associated with the package is updated to the latest revision from version control. The XMI file is imported into your model database, updating the package in your model.</p>	<p>Update All Packages [369]</p>
2	<p>When working in a Private Model, if Enterprise Architect determines that the package content in the model is already up to date with the latest revision of the package file retrieved from version control, then the Import Package dialog displays. This dialog is not displayed for Shared models.</p> <p>The following options are offered;</p> <ul style="list-style-type: none"> • Force reload From XMI • Accept current package <p>Choose Accept current package to skip the process of re-importing the package from XMI.</p> <p>The option Refresh model view allows you to specify whether or not to refresh the Project Browser and diagrams, by reloading the package content from the project database.</p> <p>The Import Package dialog also has an option Always use these settings. When selected, if you subsequently check out a package that is found to be up to date, the same settings are applied again without displaying the dialog.</p> <ul style="list-style-type: none"> • To display the dialog if, for example, you want to change the settings, press a Ctrl key while you select the Package Control Get Package menu option. <p>The package file is imported into your model, in accordance with your selection.</p>	

Notes

- The **Get Latest** command is disabled for any package that is checked-out (to anybody) in the currently loaded project.
- When using a Shared Model environment, where all users are connected to a single model database, you should reload the package from the database, rather than using the Get Latest command.

3.5.1.4.10 Update to the Latest Revision of All Packages

When you are part of a team working in a Distributed Model environment, from time to time, you will want to update your model with the changes that other team members have committed into version control. The command Get All Latest, allows you to bring in updates made by other users, for all of the version controlled packages in the currently loaded project.

Use to

- Update all packages in your project, to the latest revision retrieved from version control

How to

Step	Action	See also
1	<p>In the Project Browser, right-click on any package and select Package Control Get All Latest.</p> <p>All of the local XML files for all of the version control configurations used in the project, are updated to the latest revision from version control.</p> <p>Enterprise Architect scans the packages in the model, to determine which ones are up to date and which are not, compared to the latest revisions of the associated package files.</p> <p>You are then prompted to choose one of the following import options for packages that are up to date;</p> <ul style="list-style-type: none"> • Import changed files only, • Always import • Prompt for each file. 	<p>Update Selected Package^[368]</p> <p>See the note below.</p>
2	<p>Click on the OK button.</p> <p>The version controlled packages in your project are updated as appropriate.</p>	

Notes

- There is no need to re-import packages that are already up to date - re-importing packages involves first deleting them from the project and then re-importing them from the XML file, which is quite time consuming as well as quite unnecessary; we strongly recommend using the default option **Import changed files only**
- The **Get All Latest** command does not update any package that is checked-out (to anybody) in the currently loaded project; otherwise, any changes not yet committed to version control would be

discarded

- When using a Shared model environment, where all users are connected to a single model database, there is no need to use the **Get All Latest** option - the information in the model database is always ahead of what is committed into version control; you should from time to time refresh your view of the model database, by reloading diagrams or reloading package content in the Project Browser

3.5.1.4.11 Include Other Users' Packages

You can retrieve packages from version control that have been created by other users, or by you in another model, and import them into your current model.

Other users might be creating packages to use in your model. If you are not sharing a SQL database or .eap file, those packages do not automatically become part of your model. If the packages have been placed into version control, you can retrieve them and import them into your model as children of an existing package, using the **Get Package** command.

Use to

- Retrieve and import packages from version control, into the current project

How to

Step	Action	See also
1	In the Project Browser, right-click on the package to use as the parent of the incoming package.	
2	Select the Package Control Get Package context menu option. The Get Shared File dialog displays.	
3	Click on the drop-down arrow of the Version Control Configuration field, and select the version control configuration associated with the package to retrieve. The file list is populated with the names of files available through that configuration, for retrieval and import into your model.	
4	Select the package file to import into your model and click on the OK button. The selected package file is imported as a new child package, under the parent package selected in step 1.	

Notes

- You must have access to the package files through the version control system and you must define a Version Control Configuration through which to access those files
- The version control configuration must use the same unique ID that was originally used to add the package to version control

- XML package files associated with packages that are already part of your project, are NOT included in the list of files available for import

3.5.1.4.12 Apply Version Control To Branches

Enterprise Architect enables you to apply version control to all packages within a selected *model branch*, in a single operation. In this context, a model branch is a package that is currently selected in the **Project Browser**, and all of the packages contained within it.

Use to

- Apply version control to all packages within a selected model branch

How to

Step	Action	See also
1	Right-click on the required package and select the Add Branch to Version Control context menu option. The Apply VC to Branch dialog displays.	
2	Click on the drop-down arrow in the Version Control Configuration field and select the configuration to use.	
3	Optionally, tick the checkbox Export as Model Branch . Once the version control operation is complete Enterprise Architect creates a Model Branch file (.EAB file) for this branch.	Export Model Branch 371 Import Model Branch 372
4	Click on the OK button. Enterprise Architect creates a number of sub-folders within the version control working copy folder, then exports all of the packages within the selected model branch. Enterprise Architect generates filenames for the XML files, based on the package GUIDs.	

Notes

- The Version Control Configuration to be used in this operation must be defined within the model before selecting this command
- When invoked on the model root node, this command applies version control to every package within the model

3.5.1.4.13 Export Controlled Model Branch

Model branch files can simplify the process of exporting and importing a package hierarchies from one model to another.

You might want to export a newly created model branch from your own private copy of a model so that, for

example:

- Another user can import that branch into their own private copy of the same model
- It can be imported for inclusion as a common branch in a number of different models

Applying version control to an Enterprise Architect model can result in many XML files placed under version control. It could then be hard to locate and import the file corresponding to the root of a particular model branch. Enterprise Architect's Model Branch Files (.EAB files) overcome this problem by simplifying the retrieval of model hierarchies for use in other models.

Use to

- Create a Model Branch File to represent a package hierarchy that is stored in version control

How to

Step	Action	See also
1	Select the version controlled package to export as a model branch.	
2	Right-click on the package and choose Package Control Export as Model Branch . The Export as Model Branch dialog displays.	Import Model Branch 372
3	In the EAB Filename field, type a name for your Model Branch File. Alternatively, click on (...) and browse for the file location. Note that the package name is supplied as a default.	
4	Click OK . A branch file is created to represent the selected package. The branch file is committed to version control using the same Version Control Configuration that controls the package selected in step 1.	

Notes

- You can specify any file name, including sub-folder names, as long as the file is contained in or below the working folder of your version control configuration
- The facility is only enabled for packages that are already under version control

3.5.1.4.14 Import Controlled Model Branch

Applying version control to an Enterprise Architect model can result in many XML files placed under version control. It could then be hard to locate and import the file corresponding to the root of a particular model branch. Enterprise Architect's Model Branch files overcome this problem by simplifying the retrieval of package hierarchies for use in other models.

The Import a Model Branch command uses Enterprise Architect's Model Branch Files, of which there are few, to retrieve information about the root package file and import the model branch. The Model Branch File records information such as the name and type of the version control configuration for the selected package, and the relative filename of the version controlled XML file associated with the package.

Use to

- Retrieve a model branch created by another user in a private copy of a model, to import it into your own private copy of the same model
- Retrieve a model branch that is common in many models, for inclusion in a new model

How to

Step	Action	See also
1	<p>Ensure that you have:</p> <ul style="list-style-type: none"> • An operational version control environment that can be accessed by Enterprise Architect, and • Downloaded all of the version controlled package files and the model branch file associated with the model branch to import, into a valid and accessible working copy folder 	Version Control Product Setup ^[324]
2	Select the package into which the model branch is to be imported.	
3	<p>Right-click on the package and choose Package Control Import a Model Branch.</p> <p>The Import VC Model Branch dialog displays.</p>	
4	<p>Choose one of either:</p> <ul style="list-style-type: none"> • Use the lower portion of the Import VC Model Branch dialog to select a model branch file <p>(This is the simpler option if the associated version control configuration has already been saved in the current model; continue with step 5)</p> <ul style="list-style-type: none"> • Click on the Find a Model Branch (.EAB) file button <p>(This option is useful when you have not yet defined the version control configuration that is associated with the model branch to be imported)</p>	Manually Locating Model Branch Files ^[374]

	If choosing this option, please see the <i>Manually Locating Model Branch Files</i> topic	
5	Click the drop-down arrow in the Select a Version Control Configuration field and select the configuration A list of EAB files controlled by that configuration is displayed.	
6	Select the Model Branch File from the Select a Model Branch (.EAB) file list.	
7	Click OK . Enterprise Architect imports the root package specified in the Model Branch File and recursively imports and populates all the sub-packages contained in the root package.	

Notes

- The **Import a Model Branch** command is only enabled for packages that you (the current user) are able to edit, as the imported model branch is inserted into the model under the selected package

Learn more

- [Export a Model Branch](#) ^[37]

3.5.1.4.14.1 Manually Locating Model Branch Files

When importing a Model Branch File from version control, sometimes you do not yet have the associated version control configuration saved in the model that is receiving the import. In this situation, it is simpler to manually locate the Model Branch File and let Enterprise Architect derive the details of the configuration from the branch file you select.

Use to

- Browse the file system to locate the Model Branch File (.EAB) to import
- Let Enterprise Architect determine the details of the required version control configuration, based on information contained in the Model Branch File

Access **Right Click on a package | Package Control | Import a Model Branch: Find a Model Branch (.EAB) file**

How to

Step	Action	See also
1	Ensure that you have an operational version control environment that can	Version Control Product

	be accessed by Enterprise Architect and that you have downloaded all of the version controlled package files and the model branch file associated with the model branch to import, into a valid and accessible working copy folder.	Setup ^[324]
2	Choose the package in the Project Browser, into which the model branch is to be imported.	
3	Right-click on the package and choose Package Control Import a Model Branch . The Import VC Model Branch dialog displays.	
4	Click on the button Find a Model Branch (.EAB) file . A file selection dialog is displayed.	Import Controlled Model Branch ^[372]
5	Browse for and select the Model Branch File that represents the model branch to import.	
6	Click Open . If the version control configuration referenced by the file is fully defined within the current model, the import commences at this point. Otherwise, Enterprise Architect displays a dialog prompting you to complete the required configuration.	
7	Click Yes to proceed with completing the definition of the version control configuration. The Version Control Settings dialog is displayed.	Version Control Settings dialog ^[346]
8	Complete the definition of the configuration. (Typically this involves simply specifying the working copy folder.)	
9	Click on the Save button. The configuration details are saved. The import of the model branch proceeds.	

Notes

- The Import a Model Branch command is only enabled for packages that you (the current user) are able to edit, as the imported model branch is inserted into the model under the selected package
- It is not necessary to have the relevant Version Control Configuration set up within Enterprise Architect before issuing this command - Enterprise Architect prompts you to complete specification of the

configuration if necessary

Learn more

- [Export a Model Branch](#)^[37]

3.5.1.4.15 Review Package History

The File History command allows you to review the change history of version controlled packages.

It also enables you to check out a prior revision of the package for editing, effectively allowing you to roll-back to a prior revision of the package.

Use to

- Review the change history of version controlled packages
- Revert to a previous revision of a version controlled package

How to

Step	Action	See also
1	In the Project Browser, select the package for which to view change history.	
2	Right-click on the package and choose Package Control File History . For version control environments using Subversion, CVS or TFS command line clients, Enterprise Architect's File Version History dialog displays. For version control environments using SCC based clients, your particular product opens its own File Version History dialog.	Review Package History - SCC Client ^[37]
3	Click on a revision number In the Revisions field, to select that revision and view its log entry.	
4	Optionally, click on either; <ul style="list-style-type: none"> • Check Out • Retrieve <p>If you click on Check Out, the selected revision of the package file is retrieved from version control and imported into your model as a package that is checked out for editing. You can subsequently check in this revision as a new HEAD revision - effectively allowing you to revert a package to a prior revision.</p> <p>If you click on Retrieve, the selected revision of the package file is retrieved from version control and imported into your model, but the package remains flagged as checked-in, so it cannot be modified. Subsequently checking out the package updates the package to the latest revision before it is unlocked for editing.</p>	

Notes

- If the selected package was already checked out in the current model, the **Retrieve** and **Check Out** buttons are disabled
- If you check out a prior revision of a package, but do not want to commit it as a new revision, right-click on the package and select **Package Control | Undo Check Out**

3.5.1.4.15.1 Review Package History - SCC Client

The **File History** command allows you to review the change history of version controlled packages. It also enables you to check out a prior revision of the package for editing, effectively allowing you to roll-back to a prior revision of the package.

The process for reviewing the package history of packages configured for version control with an SCC client (which includes products such as Visual Source Safe, TFS-SCC, ClearCase, Perforce, AccuRev, MKS Source Integrity and Serena Dimensions) differs from that for Subversion, CVS or TFS command line clients.

Use to

- Review the change history of version controlled packages
- Revert to a previous revision of a version controlled package

How to

Step	Action	See also
1	In the Project Browser, select the package for which to view change history.	
2	Right-click on the package and choose Package Control File History . The change history mechanism offered by the third party SCC provider displays.	
3	To import a prior revision of the package into your model, use the SCC History dialog to retrieve the revision, then close the dialog. The SCC client notifies Enterprise Architect that a different revision has been retrieved. Enterprise Architect then displays a prompt, asking whether to check-out the prior revision.	Example - Retrieve Prior Revision From SCC ^[378]
4	Optionally, click on either; <ul style="list-style-type: none"> • Yes, to check out the prior revision • No, to import the prior revision as read-only <p>If you click on Yes, the selected revision of the package file is retrieved from version control and imported into your model as a package that is checked out for editing. You can subsequently check in this revision as a new HEAD revision - effectively allowing you to revert a package to a prior revision.</p>	

	<p>If you click on No, the selected revision of the package file is retrieved from version control and imported into your model, but the package remains flagged as checked-in, so it cannot be modified. Subsequently checking out the package updates the package to the latest revision before it is unlocked for editing.</p>	
--	--	--

Notes

- If the selected package was already checked out in the current model, Enterprise Architect does not proceed with retrieving a prior revision
- If you check out a prior revision of a package, but do not want to commit it as a new revision, right-click on the package and select **Package Control | Undo Check Out**

Depending on your version control product, retrieving the prior revision can involve a number of prompts regarding overwriting the current local copy.

The following example details retrieval of a prior revision from a TFS-SCC version control configuration.

How to

Step	Action	See also
1	Display the TFS File History dialog.	Package History - SCC [377]
2	<p>Click on the Get button.</p> <p>The TFS-SCC client displays the Resolve Conflicts dialog.</p> <p>This dialog offers an option to Automerge All XML package files. DO NOT select this option.</p> <p>It is important to prevent any merging of Enterprise Architect's XML package files.</p>	
3	<p>Click on the Resolve button.</p> <p>The TFS-SCC client displays the Resolve writable file conflict dialog.</p>	
4	<p>Select the Overwrite local file/folder option.</p> <p>The existing working copy of the package file is overwritten by the prior revision retrieved from version control.</p>	
5	<p>Click the OK button.</p> <p>The TFS-SCC client redisplay the Resolve writable file conflict dialog; it should now show no conflicts.</p>	
6	Click on the Close button.	

	The TFS-SCC client redisplay the file History dialog.	
7	Click on the Close button. Enterprise Architect displays a prompt, asking whether to check-out the prior revision.	
8	Click on the: <ul style="list-style-type: none"> • Yes button to check-out the prior revision • No button to retrieve a read-only version of the package, that is NOT checked-out and is NOT editable 	

3.5.1.4.16 Validate Package Configurations

You can test the validity of the version control settings associated with each version controlled package within your current model.

Access [Project | Version Control | Validate Package Configurations](#)

How to

Step	Action	See also
1	The validation process scans the model database and verifies that the version control configuration associated with each version controlled package is fully specified in the current model. It also queries the corresponding version control provider to find the status of the package file associated with each version controlled package. The results of the validation process are sent to the Enterprise Architect Output window.	
2	Open the Version Control Settings dialog to complete the definition of any invalid or missing version control configurations.	Version Control Setup ^[344]
3	Click on an error message in the Output window to highlight the corresponding package in the Project Browser.	
4	Right-click a package node and choose Package Control Configure Package to open the Package Control Options dialog. Correct any problems with the version control details for the package. Correct any problems with the package's associated XMI file.	Configure Controlled Package ^[362]

3.5.1.4.17 Resynchronize the Status of Version Controlled Packages

Enterprise Architect enables you to re-synchronize the version control status of version controlled packages within your project with the status reported by your version control provider. This can be useful if you are creating copies of your Enterprise Architect project, where checking in a package from one copy of the model leaves the package in the second copy of the model with an out of date version control status.

For a given package, the re-synchronization process queries the corresponding version control provider to find the status of the package file associated with the version-controlled package. If necessary, the process then updates the package flags within the model database, to synchronize the package status recorded in the model with the value reported by the version control provider.

Use to

- Update the version control status of packages in your model, to align it with the status reported by their associated Version Control Configuration

How to

Step	Action	See also
1	<p>You can resynchronize either a single version controlled package or all version controlled packages within your project.</p> <p>Either;</p> <ul style="list-style-type: none"> • Right-click the package in the Project Browser and select the Package Control Re-synch Status With VC Provider context menu option, or • Select the Project Version Control Re-synch Statuses of All Packages option <p>The results of the re-synchronization process are sent to the Enterprise Architect Output window.</p>	
2	Double-click on any result message to select, in the Project Browser, the corresponding package.	

Notes

- This process does not cause any package data to be either exported from your model to the associated package file, or imported from a package file into your model's package data
- If a package has been checked-out and modified with Enterprise Architect, but your version control provider reports the package file as checked-in, running this process marks the package within Enterprise Architect as being checked-in, without exporting and committing the pending changes; subsequently checking-out the package imports the latest revision of the package file from version control, effectively discarding the uncommitted modifications from the model
- Similarly, if a package file is checked-out to you in your local working copy folder, but not in the Enterprise Architect model, running this process marks the package within the model as checked-out,

but it does not import the associated package file from the version control system; consequently, it is possible to check-in a package from Enterprise Architect that is potentially out of date, compared to the latest revision of the package file within the version control system

3.5.2 Tracking Changes

Enterprise Architect provides two separate but complementary facilities for tracking changes to data across the project.

Topics

Topic	Detail	See also
Auditing of model changes	<p><i>Auditing</i> is a project-level feature, available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions, that enables you to record model changes in Enterprise Architect.</p> <p>By enabling this feature, model administrators can view a range of information regarding changes, such as:</p> <ul style="list-style-type: none"> • Who changed an element • How many elements they changed • When they changed the data • What the previous values were, and • What type of elements they changed 	Auditing ^[382]
Baselining and differencing to capture and roll back changes	<p>The Enterprise Architect Corporate, Business and Software Engineering, System Engineering and Ultimate editions provide a facility to <i>baseline</i> or snapshot a model branch in XML format at a particular point in time, and store it within the model in compressed format.</p> <p>More than one baseline can be stored against a single Enterprise Architect package; using baselines, you can compare packages at the current and earlier stages of development, using the <i>Compare (Diff)</i> utility.</p> <p>The Compare utility is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect; it enables you to compare the current model with:</p> <ul style="list-style-type: none"> • A Baseline • An exported Enterprise Architect XML file on disk • A version-controlled Enterprise Architect XML file on disk 	Package Baselines ^[394] The Compare Utility (Diff) ^[401]

3.5.2.1 Auditing

Auditing is a project-level feature that enables model administrators to record model changes in Enterprise Architect.

Guide:

Topic	Detail	See also
Overview	<p>By enabling Auditing, you can view information on changes such as:</p> <ul style="list-style-type: none"> • Who changed an element • How many elements they changed • When they changed the data • What the previous values were, and • What type of elements they changed <p>Auditing does not record changes to:</p> <ul style="list-style-type: none"> • RTF Templates • Model Documents • Baselines • Profiles 	
Quickstart	You can quickly enable auditing and see it in action.	Auditing Quickstart ^[383]
Settings	Once auditing is enabled within a project, you have a variety of options available for customizing what is recorded by the audit.	Audit Settings ^[384]
The Audit View	<p>To view what has been recorded by the audit, use the Audit View, which provides an interface to everything recorded by auditing.</p> <p>If security is enabled, you must have Audit View permission to display data in the Audit View.</p>	Audit View ^[387] List of Available Permissions ^[264]
Model Views	<p>You can also obtain a snapshot of selected items in the model, using the <i>Model View</i> facility.</p> <p>In the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions of Enterprise Architect, this facility enables you to automatically generate a snapshot at intervals and, if there are changes in the items collected by the defined search, to trigger a notification to you of such changes.</p> <p>This enables you to monitor work flow and other events of concern to you.</p>	Model View ^[598] Monitor Events ^[532]

Topic	Detail	See also
RTF Report	You can generate an RTF report that includes the audit history information for the selected element or package, by choosing the <i>basic + audit</i> RTF template.	RTF Dialog Options ^[2256]
Audit History	Using Auditing, you can track changes to an element or connector over time. However, enabling Auditing also enables an Audit History tab in the System Output window, which summarizes all changes made to the selected element or connector.	Audit History Tab ^[392]
Performance Issues	By enabling auditing on a project, you increase the time taken for most actions. For most modeling tasks, this increase is insignificant, however, there are some instances where the difference is more substantial.	Performance Issues ^[393]

Notes

- The Auditing facility is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Warning** - If your site runs separate editions of Enterprise Architect, when Auditing is turned on in a project any Desktop or Professional edition users are locked out of the project; to restore access, turn Auditing off in the project from the Corporate, Business and Software Engineering, Systems Engineering or Ultimate edition instance of Enterprise Architect

3.5.2.1.1 Auditing Quickstart

You can quickly enable auditing and see it in action.

Access **Project | Audit View: Audit Settings button**

How to

Step	Action	See also
1	Select the Audit View menu option to open the Audit View.	The Audit View ^[387]
2	Click on the Audit Settings button. The Audit Settings dialog displays.	Auditing Settings ^[384]

Step	Action	See also
3	Select the Enable Auditing checkbox.	
4	Click on the OK button to close the Audit Settings dialog.	
5	Close the Audit View dialog.	
6	Change and save your project <ul style="list-style-type: none"> • Add a new package • Add a new Class • Add a new connector • Change the name of an element • Delete an element 	
7	Open the Audit View again and click on the Refresh (or Load) button to display a record of the changes you made.	

3.5.2.1.2 Auditing Settings


The Audit Settings dialog enables you to change what is recorded by the Auditing facility.

Access [Project | Audit View: Audit Settings](#)

Use to

- Define the areas of processing in Enterprise Architect to audit
- Administer your audit records
- Indicate the kind of model objects for which changes are to be recorded
- Configure auditing to record changes to only certain types of elements

Reference

Action	Usage	See also
Enable Auditing	Turn the Auditing facility on.	
Audit XMI Import	Include XMI importing in the audit. As version control uses XMI, this option must be selected to record changes from checking out packages.	Version Control 

Action	Usage	See also
Audit XMI Export	<p>Include XMI exporting in the audit.</p> <p>As version control uses XMI, this option must be selected to record changes from checking out packages.</p>	Version Control ^[312]
Audit Reverse Engineering	<p>Include reverse engineering in the audit.</p>	
Use Database Timestamp	<p>Use the database server's timestamp instead of each user's local timestamp; this improves security.</p> <p>This option is not available for .eap project files.</p>	
Save Logs	<p>Save a copy of the logged audit items currently held in the project.</p> <p>These items remain in the project; use Clear Logs to remove them.</p> <p>Enterprise Architect prompts you to specify whether to save items covering a specific period of time.</p> <ul style="list-style-type: none"> • Click on the No button to save all log items currently held in the database • Click on the Yes button to display the Time Filter dialog, on which you select a standard time period or define your own <p>This function can be accessed through the Automation Interface.</p>	Audit View Controls ^[389] Repository ^[2448]
Clear Logs	<p>Permanently delete all log data from the current project; use the Save Logs button first, to save the audit records outside the project.</p> <p>Enterprise Architect prompts you to specify whether to clear items covering a specific period of time.</p> <ul style="list-style-type: none"> • Click on the No button to clear all log items currently held in the database • Click on the Yes button to display the Time Filter dialog, on which you select a standard time period or define your own <p>This function can be accessed through the Automation Interface.</p>	Audit View Controls ^[389] Repository ^[2448]
Load Logs	<p>Load a previously saved set of logs back into the project;</p>	Repository ^[2448]

Action	Usage	See also
	<p>if the same record exists in both project and file, it is not reloaded.</p> <p>This function can be accessed through the Automation Interface.</p>	
Core	Record changes to elements (including attributes and operations), packages, connectors and some model-level information.	
Standard	Record the same changes as the Core option, plus changes to diagrams.	
Extended	Record the same changes as the Standard option, plus changes to security.	
Maintenance	<p>Audit maintenance elements only; that is:</p> <ul style="list-style-type: none"> • Package • Requirement • Feature • Use Case • Actor • Note • Issue • Change 	
Core Structural	<p>Audit maintenance elements plus some structural elements; that is:</p> <ul style="list-style-type: none"> • Package • Class • Interface • Signal • Node • Component • Artifact • Part • Port • Device 	

Action	Usage	See also
All	Audit all elements.	
Custom	<p>Audit element types that you specify.</p> <p>The Customize button is made available; click on this button to display a list of element types, and select the checkbox against each element type to include in the audit (or click on the Select All button to select every element type).</p> <p>Click on the OK button to save the selection.</p>	

Notes

- As the number of records increases, the performance of the Audit View reduces; it is recommended that audit records that are not regularly required are saved to file, then cleared from the project, to help ensure high performance
- Connectors are audited when they are connected to an element that is included in the Audit Options
- If security is enabled, you must have **Audit View** permission to turn Auditing on, and **Audit Settings** permission to change Audit settings

3.5.2.1.3 The Audit View

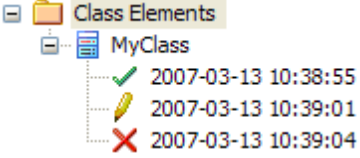
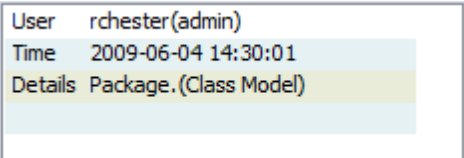
The Audit View provides an interface to the information that has been recorded by auditing.


Access **Project | Audit View**

Guide:

The Audit View is divided into three main areas:

Action	Usage	See also
View controls	The view controls provide a variety of settings for controlling auditing and the display of audit records.	Audit View Controls ^[389]
Audit tree	<p>The audit tree displays the log items that have been recorded by auditing.</p> <p>What is displayed in the tree is affected by the view controls, such as:</p> <ul style="list-style-type: none"> • Sorting • Filter (by time) • Mode • Auditing settings (what was actually recorded) 	Audit View Controls ^[389] Auditing settings ^[384] The Project Browser ^[567]

Action	Usage	See also
	<p>If synchronizing with the Project Browser, it is also affected by the package, diagram or element you have selected.</p>  <p>In the audit tree:</p> <ul style="list-style-type: none"> • The green tick indicates a creation • The yellow pencil indicates an edit • The red cross indicates a deletion <p>Right-clicking an element in the audit tree (such as <i>MyClass</i>) displays a context menu that enables you to locate the selected element in:</p> <ul style="list-style-type: none"> • The Project Browser • Any diagrams in which it exists 	
<p>Record display</p>	<p>The record display is in two parts: the identity of the selected change, and the actual change made.</p> <p>The data in the record display is determined by the view controls and mode and, if synchronizing with the Project Browser, by the package, diagram or element you have selected.</p> <p>Identity:</p>  <p>The identity of a change consists of:</p> <ul style="list-style-type: none"> • The Windows username of the user that made the change; if security is enabled, the name is of the format <i>WindowsUsername(SecurityUser)</i> • When the change was made • The path of the change; for example: Class <i>Class1</i> - Attribute <i>Att1</i> - Attribute Constraint <i>Constraint</i> 	<p>Audit View Controls^[389]</p> <p>The Project Browser^[567]</p>

Action	Usage	See also
	<p>Changes:</p> <p>Changes are displayed in a table format, showing the:</p> <ul style="list-style-type: none"> • Property (or data item) name • Its original value before the change and • Its value after the change <p>If you double-click on an item in the list of changes (or right-click and select the Show Differences context menu option) the Difference window displays:</p>  <p>This shows the specific changes that have been made, highlighting the edited, created, deleted or formatted characters.</p> <p>Changes to the <i>format</i> of text in the element, made through the element Properties dialog, are not apparent in the initial table; they are visible in the Difference window, identified by HTML formatting tags.</p>	

Notes

- If security is enabled, you must have **Audit View** permission to display data in the Audit View

Learn more

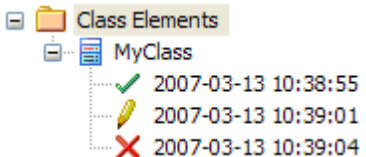
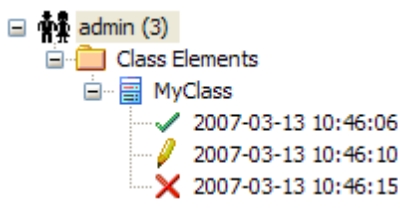
- [List of Permissions](#) ²⁶⁴

3.5.2.1.3.1 Audit View Controls

The **Audit View** controls provide a variety of settings for controlling auditing and the display of audit records.

Reference

Action	Usage	See also
Load or Refresh	Click to reload the Audit Tree, updated with any new audit results.	
Search	Click to search through log items for a particular area - you can search by Name, Type or GUID.	

Action	Usage	See also
	<p>The items are loaded and filtered with the current Sort By, Time Filter and Mode settings.</p> <p>If you refresh the Audit View, you must run the search again.</p>	
Audit Settings	Click to open the Audit Settings dialog.	Audit Settings ^[384]
Sort-by	<p>Select the appropriate radio button for the required display setting:</p> <ul style="list-style-type: none"> Type - changes are grouped under element type (such as Class or Requirement), and then grouped under the changed element  <ul style="list-style-type: none"> User - changes are grouped under user name, each with the number of changes for that user <p>Under each user name, changes are grouped as for the Type sort</p> 	
Filter By Date/Time	<p>Select to enable the Filter Settings button, to filter the audit results by time period.</p> <p>Changes that occur <i>outside</i> the selected filter period are not shown in the Audit View.</p>	
Filter Settings	<p>Click to display the Time Filter dialog, to set the filter time period:</p> <ul style="list-style-type: none"> Today - to display changes occurring today Previous Hour - to display changes occurring in the last 60 minutes Previous 24 Hours - to display changes occurring in 	

Action	Usage	See also
	<p>the last 24 hours</p> <ul style="list-style-type: none"> • Previous Week - to display changes occurring in the last 7 days • Previous 30 Days - to display changes occurring in the last 30 days • Previous Year - to display changes occurring in the last 365 days • Custom - to define your own time period, using the From and To fields <p>The six pre-configured time periods automatically update when you click on the Refresh button; custom periods are static and do not automatically update.</p> <p>If you have set a filter period, and you deselect the Filter By Date/Time checkbox, the period remains set; the custom time period, too, is retained so that you can re-use it or modify it later if required.</p>	
Status Text	<p>Read to see which:</p> <ul style="list-style-type: none"> • Mode has been selected and • Time filter is being applied to the data 	
Mode	<p>Click to display a menu of options to change the mode of the Audit View.</p> <p>Select:</p> <ul style="list-style-type: none"> • Standard - to automatically synchronize with the Project Browser; where changes have been made, the Audit View reflects your selection from the Project Browser - if you click on: <ul style="list-style-type: none"> • An element, the Audit View displays the history for that element • A package, the Audit View displays the history for that package and its immediate children (but not the contents of nested packages) • A diagram, the Audit View displays the history for that diagram and its contents (which could be drawn from a wide area of the Project Browser) • Advanced - to load large sets of log items independent of the Project Browser; in this mode, a 	The Audit View ³⁸⁷

Action	Usage	See also
	<p>special Audit Settings group can be displayed in the Audit Tree, which records:</p> <ul style="list-style-type: none"> • When Auditing has been enabled and disabled • Who made the change • The date and time of the change • Changes to the Audit Settings • When Audit Operations are executed • Security changes (which can be browsed in the same way as other changes) <ul style="list-style-type: none"> • Deleted - to display only deleted records, but otherwise data is shown as in Advanced mode; records can be sorted by element type or by user as required • Raw - to display all audit records in chronological order without sorting, enabling you to see a progression of changes; this can be especially useful in determining date-time inconsistencies <p>Any search and filtering you define still apply, enabling you to view all of today's changes in order, or all changes for a particular element in order, or both</p> <p>Additional database information is displayed; this additional information might be insignificant or only in machine-readable format</p>	

3.5.2.1.4 Audit History Tab

When Auditing is turned on, an Audit History tab is enabled in the System Output window.

Access [View | System Output > Audit History tab](#)

Guide:

Topic	Detail	See also
Prerequisite	To see this tab, you must have the Audit View open.	The Audit View ³⁸⁷

Topic	Detail	See also
	The tab continues to display if you subsequently close the Audit View.	
Information Shown	<p>The tab shows a history of changes to whichever element or connector you have selected in the:</p> <ul style="list-style-type: none"> • Current diagram • Project Browser • Audit View, or • Package Browser <p>As you select different elements or connectors, the Audit History tab automatically updates to reflect your current selection.</p> <p>For each change made to the element or connector, the tab shows:</p> <ul style="list-style-type: none"> • Who made the change • When the change was made • Where the change was made • The value of the characteristic before the change • The value of the characteristic after the change 	

Notes

- If security is enabled, you must have **Audit View** permission to display data on the Audit History tab

Learn more

- [System Output Window](#)^[164]
- [List of Permissions](#)^[264]

3.5.2.1.5 Auditing Performance Issues

Enabling auditing on a project increases the time taken for most actions.

For most modeling tasks, this increase is insignificant; however, there are some situations where the difference is more substantial.

Topic	Detail	See also
Large Deletions	<p>Deleting large packages or package structures, or large numbers of elements, takes significantly longer with auditing on.</p> <p>You might disable auditing before performing such a deletion.</p>	Auditing Settings ^[384]

Topic	Detail	See also
XMI Imports	Importing XMI takes longer with auditing enabled. A project-level option is provided for disabling auditing of XMI Imports.	Import From XMI ^[417] Auditing Settings ^[384]
Reverse Engineering	Reverse engineering code takes longer with auditing enabled. A project-level option is provided for disabling auditing of reverse engineering.	Import Source Code ^[1790] Auditing Settings ^[384]
Replication	You cannot remove replication from a model with Auditing enabled. If you have to remove replication, disable Auditing and - if prompted to do so - allow Enterprise Architect to roll back the database version; you can then remove replication.	Remove Replication ^[248] Auditing Settings ^[384]

3.5.2.1.6 Audit View Performance Issues

Most operations in the Audit View are affected by the volume of use of the database - both by other facilities and by auditing itself.

Some potential problems and their solutions are outlined below:

Topic	Detail	See also
Navigating the Project Browser Within Auditing is Slow	Try setting the time filter to a period in the immediate past, such as Today , Previous 24 Hours or Previous Week ; this time period updates each time you open or refresh the Audit View.	Audit View Controls ^[389] Auditing Settings ^[384]
The Audit View is Slow in Loading and Changing Modes	Save log items outside the project with the Save Logs button; if you then clear the logs you have just saved, the load time of the Audit View is reduced. You can reload logs into the project at any time, using the Load Logs button.	
Navigating the Audit Tree is Slow	Close the Audit History tab in the Output window	Audit History Tab ^[392]

3.5.2.2 Package Baselines

Enterprise Architect includes tools to help you manage and review changes to your models over time. These tools apply the concepts of *Baselines*, *Differencing* and *Merges*.

You use Baselines, Differencing and Merges essentially to compare two snapshots of a specific part of your project, to capture the differences between them and either roll back or incorporate selected changes or all

changes.

Topic	Detail	See also
Baselines	<p>Enterprise Architect provides a facility to create a Baseline or 'snapshot' of the contents of a selected package and its child packages at a particular point in time; this enables you to later compare that branch of the model at that time with the current state of the branch.</p> <p>Baselines are stored in the same XML format as is used for version control, but are stored within the project in compressed format.</p> <p>You can also have parallel copies of parts of your model for team development, and create Baselines within each copy to merge changes into the project master.</p>	Baselines ^[396]
Differencing	<p>Differencing (<i>Diff</i>, or <i>Compare</i>) enables you to explore the differences between:</p> <ul style="list-style-type: none"> • The current state of a specific part of your project, and • Previous or parallel versions captured in a Baseline or an XMI 1.1 file on disk <p>The section includes an example of such a comparison.</p>	The Compare Utility (Diff) ^[401] Example Comparison ^[406]
Merges	<p>Once Differencing is complete, you can merge information from the Baseline into the current project; it is not possible to go the other way.</p> <p>You can:</p> <ul style="list-style-type: none"> • Merge information manually, change by change • Merge information automatically by electing to merge in all changes in one batch procedure • Revert completely to the original Baseline by importing the stored XMI directly • Merge information and elements from a Baseline in a different project, making it possible to keep multiple versions of a single model in synch <p>The merge options are available through the toolbar, context menus and the keyboard on the Compare Utility tab, which shows the results of a comparison.</p>	Compare Utility Tab Options ^[408]
Visual Differences in Diagrams	<p>Changes to a model might include:</p> <ul style="list-style-type: none"> • Adding or removing elements and connectors on a 	Check Visual Changes to Diagrams ^[403]

Topic	Detail	See also
	<p style="text-align: center;">diagram, or</p> <ul style="list-style-type: none"> • Changing the position of elements or the overall layout of a diagram <p>You might believe that a diagram has changed, and select to compare it with a baseline using a context menu option from the Project Browser. Alternatively, you might perform a baseline comparison on a package or a model and select from the comparison output any diagrams that are flagged as changed.</p>	

Notes

- Package Baseline facilities are available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect
- The Enterprise Architect Corporate, Business and Software Engineering, System Engineering and Ultimate editions provide another facility, Auditing, which you can switch on to perform continuous monitoring of changes across the project; you can dovetail your use of each facility to meet the range of your change management requirements
- If a package under version control forms part of a Baseline, and that package is checked in to the model, you cannot merge the original data from the Baseline into that package
- You can also obtain a snapshot of selected items in the model, using the Model Views facility; this facility enables you to automatically generate the snapshot at intervals and, if there are changes in the items collected by the defined search, to trigger a notification to you of such changes, which enables you to monitor work flow and other events of concern to you
- If security is enabled you must have **Baselines - Manage** permission to create, import and delete Baselines, and **Baselines - Restore** permission to merge data from a Baseline; security permissions are not required to select an existing Baseline and perform a comparison with the model

Learn more

- [Auditing](#) ^[382]
- [List of Permissions](#) ^[264]
- [Model Views](#) ^[598]
- [Monitor Change Events](#) ^[532]

3.5.2.2.1 Baselines

Enterprise Architect provides a facility to 'Baseline' (snapshot) a model branch at a particular point in time for later comparison with the current package state.

Topic	Detail	See also
Baselines	Baseline comparison is most useful for determining the	The Compare Utility

Topic	Detail	See also
	<p>changes made to the model during development compared to some Baseline saved at a crucial point - for example the completion of a phase or version iteration.</p> <p>More than one Baseline can be stored against a single Enterprise Architect package.</p> <p>Baselines are particularly useful during requirements management to check for changes, additions and deletions that have occurred since the start of the current work phase; knowing how a model has changed is an important part of managing change and the overall development process.</p> <p>Baselines are stored within the model in compressed XML format; you can save a Baseline to an external XML file for storage or archive, or for distributing to other users working on models derived from a master project.</p> <p>Baselines are generally used in conjunction with the <i>Compare</i> utility.</p>	(Diff) ^[401]
Scenario	<p>A typical scenario for using Baselines would be:</p> <ul style="list-style-type: none"> • Create the base model branch to a sufficient point to create a Baseline (checkpoint); create and store the Baseline as Version 0.1a • As work continues on development, managers and developers can check the current model branch against the Baseline for important modifications, additions and deletions; the Compare utility can be invoked from the Baseline dialog to check the current model branch against the stored version • As required, minor Baselines can be created to check recent progress; these 'temporary Baselines' are useful for managing change when a lot of work is being done and it is important to only see what has changed in, for example, the last 24 hours • At sign-off or the move to a new version/phase, a major Baseline can be created to capture the new state of the model <p>Minor Baselines created earlier can be deleted if required to save space</p>	Manage Baselines ^[398] Create Baselines ^[400]
Considerations	<ul style="list-style-type: none"> • Baselines are based on the GUID or unique ID of a particular package: 	

Topic	Detail	See also
	<ul style="list-style-type: none"> • Enterprise Architect checks for that ID as the root element within the XML document being used as a Baseline • When you export a package to XML, the package you export is the root element; likewise when you create a Baseline, the current package is the root package of the XML Baseline • When you save information in a version control system, the current version-controlled package is again the root package of the document • It is not useful to create a Baseline by importing an XML package file created by version controlling a package that itself contains version-controlled child packages; that type of XML package file contains stubs for the child packages, not full information on the child packages and elements • If a package under version control forms part of a Baseline, and that package is checked in to the model, you cannot merge the original data from the Baseline into that package • XML files must be in the same format used by the Baseline engine - currently the UML 1.3 XMI 1.1 format (plus Enterprise Architect extensions), which contains all the information necessary to reconstruct a UML model, even a UML 2.x model 	

Notes

- The Baseline facility is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- The Compare utility is available in the *Professional* edition of Enterprise Architect, as well as in the Corporate and extended editions above

3.5.2.2.1.1 Manage Baselines

Enterprise Architect provides a range of facilities for working with and managing Baselines.

Access Right-click model branch package | Package Control | Manage Baselines

Reference

Create, select and process Baselines using the Manage Baselines dialog, as indicated below:

Field	Usage	See also
Current Baselines For Package: <Name>	<p>Review the Baselines for the current model branch, listed by version reference with the highest alphabetical/numerical value at the top.</p> <p>If an entry is longer than the display area, a horizontal scroll bar displays at the bottom of the panel; use this to scroll to the text that is not shown.</p>	Create Baselines [400]
Show Differences	<p>Run the Compare [401] utility on the selected Baseline and the current model branch or diagram, to display the differences [406] between the two.</p>	The Compare Utility [401] Check Visual Changes to Diagrams [403] Example Comparison [406]
Restore to Baseline	<p>Completely restore the model branch from the selected Baseline.</p>	
New Baseline	<p>Create a new Baseline.</p>	Create Baselines [400]
Delete Selected	<p>Delete the selected Baseline.</p>	
Load Other Baselines	<p>Display a drop-down menu that enables you to load Baselines from another model, in either a .EAP file or a DBMS file.</p> <ul style="list-style-type: none"> For .EAP files, a browser displays; locate the required project file For DBMS files, the Windows Data Link Properties dialog displays; select the data provider and click on the OK button to display the Select Data Source dialog, from which you select the required project <p>In either case, the <i>Connected To:</i> message at the bottom of the Manage Baselines dialog changes to the name of the alternative model.</p> <p>To return the dialog to the original project, select the third option on the drop down list: Load From Selected Package.</p>	
Import File	<p>Import an XML 1.1 file from the file system as a new Baseline for</p>	

Field	Usage	See also
	this current model branch.	
Export File	Export the selected Baseline to an XML file on disk.	
Compare Model to File	Compare the selected model branch with an XML 1.1 file on disk; a browser displays, which you use to locate the file.	
Options	Set filters ^[402] to make the comparison more specific.	Compare Options ^[402]

3.5.2.2.1.2 Create Baselines

Access [Project Browser package context menu | Package Control | Manage Baselines: New Baseline](#)

Reference

Create the new baseline using the New Baseline dialog, as indicated below:

Field	Usage	See also
Name	Display the package name of the currently selected model branch.	Manage Baselines ^[398]
Version	Type a unique version reference for this Baseline, which can consist of any alphanumeric characters. The Manage Baselines dialog sorts the Baselines according to the value of this field.	
Include Sub-packages	Include the entire sub-package hierarchy of this branch in the Baseline; this option defaults to selected. If you deselect the checkbox, only the immediate contents (XML stubs) of the package are included in the Baseline.	
Note	Edit the default current time and date to any other value. The field is a single-line entry, for display on the Manage Baselines dialog (a one-line-per-entry list).	
OK	Click to create a new Baseline and return to the Manage Baselines dialog.	

3.5.2.2.1.3 The Compare Utility (Diff)

Enterprise Architect has a comprehensive and powerful built in *Compare (diff)* utility, which enables you to:

- Explore what has changed within a model over time
- Explore how previous versions of a model branch differ from what is currently in the model
- Perform a full model comparison by exporting all of Model A to XML, then using **Compare Model to File** from within the current model (Model B)

Comparing and checking model development at various points in the process is an important aspect of managing change and development, monitoring what is being modified and ensuring the development and design process is on track.

Access

- **Project Browser package context menu | Package Control | Manage Baselines: Show Differences** - use the Baseline dialog
- **Project Browser package context menu | Package Control | Compare with XMI File** (package not under version control)
- **Project Browser package context menu | Package Control | Compare with Controlled Version** (package under version control)

Use to

- Compare a model branch in Enterprise Architect with a Baseline created using the Baseline functionality (Corporate, Business and Software Engineering, System Engineering and Ultimate editions)
- Compare a model branch in Enterprise Architect with a Baseline stored in a different model
- Compare a model branch in Enterprise Architect with an XML 1.1 file on disk created previously using the Enterprise Architect XML export facility (user selects file)
- Compare a model branch in Enterprise Architect with the current version-controlled XML 1.1 file on disk as created when using Version Control in Enterprise Architect (file automatically selected)

Guide:

Topic	Detail	See also
Differencing With Baselines	<p>As a Baseline is stored within a model and contains all the information, elements and connections for a package at a point in time, it can be used within Enterprise Architect to track changes to model elements over time.</p> <p>The Differencing engine first builds a representation of the current package in memory, based on what is currently in the model.</p> <p>It then compares this with the stored Baseline, highlighting changes, new elements, missing elements and elements</p>	<p>Compare Options ^[402]</p> <p>Create Baselines ^[400]</p> <p>Example Comparison ^[406]</p>

Topic	Detail	See also
	<p>that have been moved to other packages.</p> <p>It is possible to filter the resultant output to display only one particular kind of change: for example, additions to the model.</p> <p>If a Baseline has been created to ignore child package content, a comparison between that Baseline and the model does not include any child package content in the model.</p> <p>See the example model comparison.</p>	

Notes

- This utility is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- You cannot compare the current model with an XML 2.1 file; the utility can only compare with an XML 1.1 file

3.5.2.2.1.4 Compare Options

The Compare Options dialog enables you to refine the output of the Compare utility when it compares the current model with a Baseline.

To display the dialog, either:

- Click on the **Options** button on the Manage Baselines dialog
- Click on the **Compare Options** icon on the Compare Utility tab toolbar

Field	Usage	See also
Always Expand to Differences	<p>Always display the list of elements fully expanded to show changes.</p> <p>If you deselect the checkbox, when the Compare Utility tab is first opened it lists the package contents to element level, and you expand each element as required to show the changed items.</p> <p>For large branches of the model, it is better to leave the checkbox unselected.</p>	
Show Elements that are	<p>List elements that:</p> <ul style="list-style-type: none"> • Have been changed since the Baseline was created • Are in the Baseline only (that is, have been deleted from the model since the Baseline was created) • Are in the model only (that is, have been created since the Baseline was created) • Have not changed since the Baseline was created (you might 	

Field	Usage	See also
	generally leave this checkbox unselected)	
Suppress these Changes	Exclude: <ul style="list-style-type: none"> • Changes to diagrams • Changes to the Date Modified field for an item • Changes to the Date Created field for an item • Child items of a deleted item • Changes to advanced properties (defaults to selected) 	
Baseline Diagram Compare Options	Select the checkbox to always open the first parent package for which there is a Baseline, when you select the diagram for comparison from the Project Browser.	

If the Compare Utility tab shows the results of a Baseline comparison, when you click on the **OK** button the display refreshes to refine the information according to the options you have selected.

Learn more

- [Compare Utility Tab Options](#)^[408]

3.5.2.2.1.5 Check Visual Changes to Diagrams

The Baseline Diagram Compare feature is a quick and easy way to **visually compare** a current diagram with an earlier version (a Baseline), and highlight any elements in the diagram that have been:

- Added
- Removed
- Resized or
- Moved

You can then review these changes and roll any of them back to the state they had in the Baseline.

The changes are identified on the Baseline Diagram Compare dialog and on the diagram itself. If the diagram is not already open, the compare feature also opens the diagram.

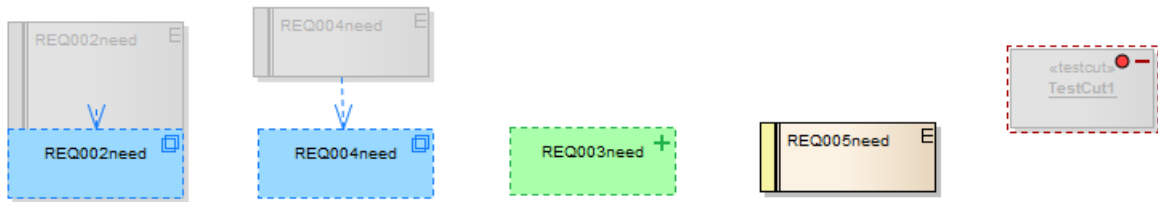
Access Any of the following:

- **Project Browser diagram context menu | Compare to Baseline | <select baseline>: Show Differences**
- **Project Browser package context menu | Package Control | Manage Baselines: Show Differences | Selected diagram context menu | Open Visual Diagram Diff**

- **Diagram context menu | Compare to Baseline: Show Differences, or**
- **Select Package | Project | Baselines: Show Differences | Selected diagram context menu | Open Visual Diagram Diff**

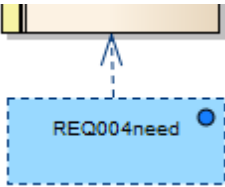
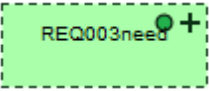

Guide:

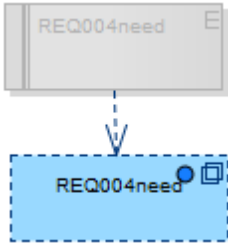
As you work on the dialog, the diagram changes as illustrated in the following diagram and as described in the table below.



Name	Status	Action
<input checked="" type="checkbox"/> REQ003need	Baseline only	Restore Element & add to Diagram
<input checked="" type="checkbox"/> REQ004need	Changed	Restore Element position
<input checked="" type="checkbox"/> REQ002need	Changed	Restore Element position
<input checked="" type="checkbox"/> TestCut 1	Model only	Delete from Diagram

Topic	Detail	See also
Select element for review	<p>The Status column indicates whether the element has been:</p> <ul style="list-style-type: none"> • Moved or re-sized (<i>Changed</i>) • Deleted from the diagram (<i>Baseline only</i>), or • Added to the diagram since the Baseline was captured (<i>Model only</i>) <p>Click on the item name.</p> <p>When an item is selected, the corresponding element on the diagram is highlighted as follows:</p> <ul style="list-style-type: none"> • Deleted from the diagram <div style="text-align: center;"> </div> <ul style="list-style-type: none"> • Added to the diagram <div style="text-align: center;"> </div>	

Topic	Detail	See also
	<ul style="list-style-type: none"> Resized or moved to a new position  <p>The currently-selected element on the diagram is marked with a dot, as shown, to indicate that it is in focus.</p>	
Position the diagram to show the selected element	<p>To scroll the diagram so that you can see the original (Baseline) position of an element, double-click on the item in the list.</p> <p>To scroll the diagram so that you can see the current (model) position of the element, press and hold (Ctrl) while you double-click on the item.</p>	
Leave the changes in the item as they are	<p>Ensure that the checkbox against the item is not selected.</p> <p>Click on the OK button.</p>	
Roll the changes back to the Baseline position	<p>Click on the checkbox against each required item (or click on the Check All button to select all items).</p> <p>The Action column displays the action required to roll each element's relationship to the diagram back to the Baseline relationship.</p> <p>The corresponding element on the diagram is highlighted as follows:</p> <ul style="list-style-type: none"> Deleted, now being restored  <ul style="list-style-type: none"> Added, now being removed  <ul style="list-style-type: none"> Resized/repositioned, now being put back to original 	

Topic	Detail	See also
	<p>position</p>  <p>The comparison shows the blue direction arrow for all repositions or resizes that have been checked. For a heavily edited diagram this can be confusing. However, you can hide the arrow for all elements except the one currently in focus; to do this:</p> <ul style="list-style-type: none"> • Deselect the Display Connector for checked items checkbox <p>To roll back the changes for all items for which a checkbox is selected:</p> <ul style="list-style-type: none"> • Click on the Apply button 	

Learn more

- [Baselines](#)^[396]
- [Manage Baselines](#)^[398]
- [Compare Options](#)^[402]
- [Compare Utility Tab Options](#)^[408]

3.5.2.2.2 Example Comparison

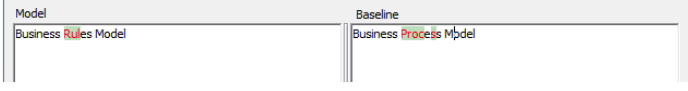
The diagram below shows the result of a comparison between a package (Business Process Model) in the current project and that package in a Baseline (version 0.1) captured at an earlier date.

The results of the comparison are displayed on the Compare Utility tab.

Model Elements	Status	Property	Model	Baseline
Business Process Model		Abstract	false	false
Rules		Alias		
People must wear safety belts in all se...	Changed	Author	Suzanne Pearson	Suzanne Pearson
No more than 5 persons in a vehicle	Changed	Date Created	16/01/2009 12:07:15 PM	16/01/2009 12:07:15 PM
Passengers under 8 years old must b...	Model only	Date Modified	5/06/2009 10:15:00 AM	16/01/2009 12:07:30 PM
Customer		Complexity	1	1
RuleFlow		Filename	C:\Documents and Settings\	C:\Documents and Settings\
Eligibility		Language	C#	C#
address	Changed	IsLeaf	false	false
Application	Moved	IsSpec	false	false
Status	Baseline only	IsRoot	false	false
Constraints		Keywords		
AllowableValues{Accept, Reje...	Baseline only	Multiplicity		
		Name	Application	Application
		Notes		
		Parent Package	Business Rules Model	Business Process Model
		Persistence		
		Phase	1.0	1.0

Guide:

Topic	Detail	See also
Interpretation	<p>A hierarchy of model elements is displayed in the left-hand pane.</p> <p>It is clearly visible, from the Status column and from the triangle-based icons, which items in the hierarchy have, since the Baseline was captured, been:</p> <ul style="list-style-type: none"> • Changed (<i>Changed</i>) • Deleted from the model (<i>Baseline only</i>) • Added to the model (<i>Model only</i>) or • Switched to different packages (<i>Moved</i>) <p>If you click on an item with a Status entry in the left hand pane, the right-hand pane displays a table of properties showing the values of those properties in the current model and in the Baseline.</p> <p>For each property where there is a difference between the model and the Baseline, the row is highlighted; for example, the Class element named <i>Application</i> was moved from the <i>Business Process Model</i> package (as shown in the Baseline column) to the <i>Business Rules Model</i> package on 5 June 2009 (as shown in the Model column).</p> <p>The Compare Utility tab enables you to perform operations (such as merging or rolling back changes) on the reported information, using the toolbar, context menu and keyboard.</p>	<p>Compare Utility Tab Options ⁽⁴⁰⁸⁾</p>
Increase Level of Detail	<p>The right panel of the Compare Utility tab might, for some fields, display only part of the value (such as <i>Advanced Properties</i>, above).</p> <p>It might also not be immediately obvious what a change is.</p> <p>In either case, you can double-click on the property to display full details and to highlight the exact differences; the following example shows the highlighted changes to the Parent</p>	

Topic	Detail	See also
	<p>Package property.</p> 	

3.5.2.2.1 Compare Utility Tab Options

The Compare Utility tab enables you to perform operations on the reported information, using the toolbar, context menu, Merge dialog and certain keyboard keys.

- The toolbar is at the top of the left-hand panel; the icons operate either on the comparison as a whole or on the currently-selected item in the left hand panel of the Compare Utility tab
- Each item in the hierarchy has a context menu, which you display by right-clicking on the item; the options displayed depend on the level of the item in the hierarchy
- The Merge dialog enables you to specify which changes to roll back in the model from the baseline
- You can use a selection of keyboard keys to move up and down the hierarchy, or to roll back changes

Reference - Toolbar:



Action	Usage	Shortcut	See also
Refresh	Re-run the comparison to refresh the current display.		
Merge To Model	Merge the values of the currently-selected item in the Baseline back into the model.		
Next Change	Highlight the next changed item (this skips <i>Moved</i> items).		
Previous Change	Highlight the previously-changed item.		
Expand All	Fully expand the selected item.		
Collapse All	Collapse the changed items in the selected item.		
Expand To Changed Items	Expand the selected item to show changed items only (in the event that you have selected to also show unchanged items in the comparison).		Compare options 402

Action	Usage	Shortcut	See also
Find in Project Browser	Highlight the item in the Project Browser.		
Log To XML	Log the changes to an XML file. A browser displays, on which you specify the file name and location.		
Compare Options	Display the Compare Options dialog.		Compare Options ^[402]
Manage Baselines	Display the Manage Baselines dialog.		Manage Baselines ^[398]
Help	Display the Help topic <i>Package Baselines</i> .		Package Baselines ^[394]

Reference - Context Menu:

Action	Usage	Shortcut	See also
Merge from Baseline Add from Baseline	Restore the item in the model to the Baseline state, or restore a deleted item from the Baseline.		
Delete from Model	Remove a recently-created item from the model.		
Merge From Baseline (with Options)	(For the root node of the hierarchy on the Compare Utility tab.) Display the Merge dialog (see below) which enables you to specify options for rolling back the whole model branch to the Baseline state.		
Refresh	(Object-level items.) Re-run the comparison to refresh the current display.		
Find in Project Browser	Locate and highlight the item in the Project Browser.		
Open Baseline Diagram Compare	(For a diagram listed in the comparison.) Displays the Baseline Diagram Compare dialog, showing differences in diagram content and layout.		Check Visual Changes to Diagrams ^[403]

Action	Usage	Shortcut	See also
Expand All	Fully expand the selected item.		
Expand To Changed Items	Expand the selected item to show changed items only.		
Collapse All	Collapse the changed items in the selected item.		
Log To XML	Log the changes to an XML file. A browser displays, on which you specify the file name and location.		
Compare Options	Display the Compare Options dialog.		Compare Options [402]

Reference - Merge Dialog:

Field	Usage	Shortcut	See also
Changed	Restore all changed items in the model branch to the Baseline state.		
In Baseline Only	Restore all deleted items to the model branch from the Baseline.		
In Model Only	Remove all recently-created items from the model branch.		
Moved	Restore all moved items to their original locations, as identified in the Baseline.		
Full Restore from XMI	Completely restore the model branch to the version held in the Baseline XMI 1.1 file, (using the XMI Import function). (This option automatically selects all the other options.)		

Reference - Keyboard:

- (**Ctrl+↓**) - expand and highlight the next changed item

- (**Ctrl+↑**) - expand and highlight the previous changed item
- (**Ctrl+←**) - undo the changes for a selected item (roll back to the Baseline values)

3.5.3 Model Transfer

Enterprise Architect enables you to transfer data between projects, using three different mechanisms.

Topics

Topic	Detail	See also
XMI Import and Export	<p>Packages can be exported from and imported into Enterprise Architect models.</p> <p>This greatly improves the flexibility and robustness of Enterprise Architect models by enabling Analysts and Modelers to externalize model elements in XMI for version control, distributed development, post processing and transferring packages between models.</p>	XMI Import and Export ^[411]
CSV Import and Export	<p>You can import and export information about Enterprise Architect elements in CSV format.</p> <p>This enables you to store and retrieve information from spreadsheet tools such as Microsoft Excel.</p>	CSV Import and Export ^[435]
Data Transfer	Enterprise Architect enables you to move complete models between files, and between files and repositories, row by row, table by table.	Perform a Project Data Transfer ^[442]

3.5.3.1 XMI Import and Export

XML Metadata Interchange (XMI), defined by the OMG and based on XML, is an open standard file format that enables the interchange of model information between models and tools.

Topics

Topic	Detail	See also
XMI and Enterprise Architect	<p>Enterprise Architect uses XMI as a method of importing and exporting model specifications between different UML packages, Enterprise Architect projects and other tools that support XMI.</p> <p>Enterprise Architect supports the XMI 1.1, 1.2, 2.1 and 2.4.1 specifications. XMI 1.1 has support for UML 1.3, whereas XMI 2.1 is used for UML 2.0 - 2.3.</p> <p>Enterprise Architect does not fully support the older 1.0 specification. When importing or exporting to XMI 1.0, some loss of data occurs due to the limitations of XMI 1.0.</p>	

Topic	Detail	See also
	<p>With XML, model details can be exchanged between different UML tools and other tools that are capable of using XML.</p> <p>Limited support for export to Rational Rose is provided using the Rose version of the XML 1.1 specification, as implemented by Unisys for Rational products.</p> <p>Packages can be exported from and imported into Enterprise Architect models, which greatly improves the flexibility and robustness of the models by enabling Analysts and Modelers to externalize model elements in XML for:</p> <ul style="list-style-type: none"> • Version control • Distributed development • Post processing and • Transferring packages between models <p>When performing Enterprise Architect-to-Enterprise Architect transfers, ensure that either XML version 1.1 or 2.1 is selected.</p> <p>When importing an XML file over an existing package, all information in the current package is deleted first, before importing from the XML file. Please make sure you do not have important changes that you do not want to lose before importing from XML.</p>	
XMI Tasks	<p>Tasks you might perform in importing and exporting XMI include:</p> <ul style="list-style-type: none"> • Setting XML Options - XMI import, export and package control all rely on saving and loading XML files; you can set a number of options to streamline this process • Exporting a package to XMI in XMI 2.1 (and earlier) • Importing from XMI with support for XMI 2.1 (and earlier) • Setting up controlled packages • Manually controlling a package by linking it to an XMI file • Batch exporting controlled packages • Batch importing controlled packages • Factoring in the limitations of XMI • Applying a UML Data Type Definition (DTD) 	<p>XML Specifications ⁵⁶²</p> <p>Export to XMI ⁴¹³</p> <p>Publish Model Package ⁴¹⁴</p> <p>Import from XMI ⁴¹⁷</p> <p>Controlled Packages ⁴²²</p> <p>Manual Version Control with XMI ⁴³²</p> <p>Batch XMI Export ⁴³⁰</p> <p>Batch XMI Import ⁴³¹</p> <p>Limitations of XMI ⁴²¹</p> <p>The UML DTD ⁴²²</p> <p>Import a Rhapsody Model ⁴²⁰</p>

Topic	Detail	See also
	<ul style="list-style-type: none"> Importing a model from IBM Rational Rhapsody 	

Notes

- XMI 2.1 exported by Enterprise Architect 7.0 (or later) might not be correctly imported into earlier versions of Enterprise Architect
- When performing Enterprise Architect-to-Enterprise Architect transfers, ensure that either XMI version 1.1 or 2.1 is selected

Learn more

- [XML/XMI Technology](#)

3.5.3.1.1 Export to XMI

The Export Package to XMI dialog is used to export a package to various XMI (XML-based) formats. This dialog, by default, allows exporting to either **XMI 1.1** or **XMI 2.1**. The exported XMI file contains all the information required for completely round-tripping the package within Enterprise Architect; that is, performing Enterprise Architect-to-Enterprise Architect transfers.

Access Click on a package in Project Browser, then:

Project | Model Import/Export | Export Package to XMI (Ctrl+Alt+E)
Right-click | Import/Export | Export Package to XMI file

Use to

- Move Enterprise Architect model elements between models, for distributed development, manual version control and other benefits

Reference

Field/Option/Button	Usage	See also
Root Package	Indicate the name of the selected package.	
Filename	Specify the file path where the XMI file is to be generated.	
Export Type	Option/Button	Usage
	XMI 1.1	Export the selected package to XMI 1.1.
	XMI 2.1	Export the selected package to XMI 2.1.
	Publish	Open the Publish Model Package dialog for Publish Model

	exporting to other versions of XML.	Package ^[414]
View XMI	View the exported XMI file.	
Export	Initiate an XMI Export.	
Close	Close this dialog.	
Help	Display this Help topic.	
Progress	List the progress of the XMI Export.	

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Export XMI** permission to export to XMI
- Alternate Images used in the package being exported are not included in the exported XMI file unless the **Export Alternate Images** option is selected in the XML Specifications section of the Options dialog
- The Log file for the package being exported is not generated unless the **Write Log** option is selected in the XML Specifications section of the Options dialog
- XMI 2.1 exported by Enterprise Architect 7.0 (or later) might not be correctly imported into earlier versions of Enterprise Architect

Learn more

- [Limitations of XMI](#)^[421]
- [List of Available Permissions](#)^[264]
- [Manual Version Control with XMI](#)^[432]
- [XML Specifications](#)^[562]

3.5.3.1.2 Publish Model Package

The Publish Model Package facility is used to export a package to various XML formats, including XMI 1.1, XMI 2.1, ArcGIS and BPMN 2.0 XML.

Access Click on a package in Project Browser, then:

Project | Model Publisher
Project | Model Import/Export | Export Package to XMI File: Publish
Extensions | Publish | <XML Format Type>
Right-click | Import/Export | Export Package to XMI File: Publish

Use to

- Export BPMN 2.0 models to BPMN 2.0 XML
- Export Enterprise Architect model elements to Rational Rose and other tools that implement the:
 - UML 1.3 XMI 1.1 / XMI 1.0 standard
 - UML 1.4 XMI 1.2 standard, or
 - UML 2.x XMI 2.x standard

Reference

On the Publish Model Package dialog:

Field/Option/ Button	Usage	See also
Root Package	Displays the name of the selected package.	
Filename	Type in or browse for the file path into which the XML file is to be generated.	
XML Type	<p>Select the XML/XMI version to which to export the package.</p> <ul style="list-style-type: none"> • Select UML 2.1 if you want to use XMI 2.1 in round-tripping a model • Select UML 2.0, 2.1.1, 2.1.2, 2.2, 2.3 or 2.4.1 if exporting to a tool that requires that specific version of the XMI <p>The selected version might render some of the following eight options unavailable.</p>	
Export Diagrams	Export all the diagrams in the selected package.	
Export Alternate Images	Export the alternate images used in the diagrams in the selected package.	Image Manager ⁷⁵⁵
Format XML Output	Format the output into readable XML (this takes a few more seconds at the end of the run).	
Write Log File	<p>Write a log of the export activity (recommended).</p> <p>The log file is saved to the directory into which the XML file is exported.</p>	
Use DTD	Use the XMI DTD when exporting to XMI 1.0 or XMI 1.1.	

	Setting this option validates the correctness of the model and checks that no syntactical errors have occurred.	
Exclude EA Extensions	Exclude tool-specific information from the export.	
Unisys/Rose Format	Export in Unisys/Rose compatible XML format.	
Generate Diagram Images	Generate the exported diagrams in the format you select from the Format drop-down list. The generated diagrams are saved in a package called <i>Images</i> in the directory into which the XML file is exported.	
Stylesheet	Select an XSL Stylesheet to post-process the XML content before saving to file (<i>optional</i>). You can post-process the XML content before saving the package to file, using an XSLT to convert the output to HTML, XSL, code or other versions of XML; to do this, you must import the required style sheet into the project through the Resources window.	Resource Window ^[1003]
View XML	View the exported XML file.	
Export	Initiate XML export.	
Close	Close this dialog.	
Help	Display this Help topic.	
Progress	Indicate the progress of the XML export.	

Notes

- When exporting and importing with XML 1.0 with Enterprise Architect, some loss of data occurs due to the limitations of XML 1.0
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Export XML** permission to export to XML
- When you select to apply a Data Type Definition (DTD) during an XML 1.1 export, the UML_EA.DTD file is written to the output directory into which the XML files are written (unless the UML_EA.DTD file is already present in the directory); no error is generated if the UML_EA.DTD file is not present in this directory during the XML export

Learn more

- [Limitations of XMI](#)^[421]
- [List of Available Permissions](#)^[264]
- [The Resources Window](#)^[1003]
- [Manual Version Control with XMI](#)^[432]
- [Export to BPMN 2.0 XML](#)^[1569]
- [Export to ArcGIS XML](#)^[1640]

3.5.3.1.3 Import from XMI

You can import a package from an XMI (XML-based) file in any of the following formats:

- UML 1.3 (XMI 1.0)
- UML 1.3 (XMI 1.1)
- UML 1.4 (XMI 1.2)
- UML 2.0 (XMI 2.1)
- UML 2.1 (XMI 2.1)
- MOF 1.3 (XMI 1.1)
- MOF 1.4 (XMI 1.2)

You can also import the *.emx and *.uml2 files generated by tools such as Rational Software Architect (RSA) and Rational Software Modeler (RSM) as well as XML files exported by ArcGIS.

Access Click on target package in Project Browser, then:

**Extensions | Import | <XML Format>
Project | Model Import/Export | Import Package From XMI File or
Right-click | Import/Export | Import Package From XMI**

Use to

- Move Enterprise Architect Model elements between models, for distributed development, manual version control and other benefits

Reference

On The Import Package From XMI dialog:

Field/Option/Button	Action	See also
Filename	Type the directory path and filename from which to import the XMI file.	
Import diagrams	Select the checkbox to import diagrams.	

Field/Option/Button	Action	See also
Strip GUIDs	Select the checkbox to remove Universal Identifier information from the file on import. This enables the import of a package twice into the same model; the second import requires new GUIDs to avoid element collisions.	
Write log file	Select the checkbox to write a log of import activity (recommended). The log file is saved in the directory from which the file is being imported, with the same name as that of the imported file plus the suffix <i>_import.log</i> .	
Import using single transaction	Defaults to selected. If the import encounters locking issues, or if you are importing a large XML file, deselect the checkbox to import the data items separately and identify problem items without blocking the whole import.	
Treat Imported Datatypes	If you are importing from Rose XML 1.1, click on the drop-down arrow and select the datatypes to add to the model.	
Import	Click to start the import.	
Close	Click to close the dialog.	
Help	Click to display this Help page.	
Import Progress	Indicates the progress of the import.	

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Import XMI** permission to import packages from XMI
- When you import an XML file over an existing package, ALL information in the current package is deleted first; before you import the XML file, please make sure you do not have important changes that you do not want to lose
- If you are importing an XML 1.1 file that was previously exported with a UML_EA.DTD file, the UML_EA.DTD file must be present in the directory into which the XML file is being written; an error occurs if the UML_EA.DTD file is absent

Learn more

- [List of Available Permissions](#) ^[264]
- [The Resources Window](#) ^[1003]
- [Manual Version Control with XMI](#) ^[432]
- [The UML DTD](#) ^[422]
- [Import EMX/UML2 Files](#) ^[419]
- [Import ArcGIS File](#) ^[1647]

3.5.3.1.4 Import EMX/UML2 Files

Rational Software Architect (RSA) enables you to add many UML models under a single root. These models can have cross references between them.

However, RSA cannot save the entire root as one file; it saves each UML model as a separate EMX file. This means that an EMX file with cross-references is not self-contained as it references elements in another EMX file.

In releases earlier than release 7.0, Enterprise Architect treats each EMX file as a separate model and hence does not allow for cross-references between them. From release 7.0, Enterprise Architect enables these cross-references.

You therefore have the options of:

- Importing a single EMX/UML2 file or
- Importing a group of related EMX/UML2 files together, thereby retaining the cross-references between the different files

Access Click on target package in Project Browser, then:

**Extensions | Import | EMX/UML2
Project | Model Import/Export | Import Package From XMI File: Other XML File Formats | EMX/
UML2, or
Right-click | Import/Export | Import Package From XM File: Other XML File Formats | EMX/UML2**

Use to

- Import Enterprise Architect Model elements from *.emx/*.uml2 files

Reference

On the Import Package from EMX/UML2 dialog:

Field/Option/Button	Action	See also
Directory	<p>Click on the (...) (Browse) button.</p> <p>The Select Import EMX / UML2 File(s) dialog displays, which enables you to select multiple files.</p> <p>Select the file or files (use (Ctrl) + click or (Shift) + click to select several files) and click on the Open button to return to the Import</p>	

Field/Option/Button	Action	See also
	Package from EMX/UML2 dialog.	
Selected Files	Lists the selected *.emx/*uml2 files.	
Write Log File	Select to write a log of import activity (recommended). The log file is saved in the directory from which the files are being imported, with the name <i>import.log</i> .	
Import	Click to import the *.emx/*uml2 files.	
Close	Click to close this dialog.	
Help	Click to display this Help topic.	
Import Progress	Indicate the progress of the import.	

3.5.3.1.5 Import a Rhapsody Model

Enterprise Architect enables you to import a model from IBM Rational Rhapsody.

Each Rhapsody project contains one Project file (with the extension *.rpy*), which is associated with a folder with the same name as the project file, ending in *_rpy*. For example, if the Rhapsody project file is called *CDPlayer.rpy*, it is associated with a folder called *CDPlayer_rpy*.

The associated folder contains all the project-related files. It might also contain a file named *filesTable.dat*, which lists all the files related to the Rhapsody project.

Access [Select package](#) | [Extensions](#) | [Import](#) | [Rhapsody, or Project Browser package context menu](#) | [Import/Export](#) | [Import Package From XMI File: Other XML File Formats](#) | [Rhapsody](#)

Use to

- Import UML Model elements from Rhapsody Project Files

Reference

Using the Import Model from Rhapsody dialog, import the model from Rhapsody as indicated below:

Field/Option/Button	Action	See also
Rhapsody Project	Click on the (...) (Browse) button.	

Field/Option/Button	Action	See also
	The Select Import Rhapsody Project dialog displays. Select a Rhapsody Project file, then click the Open button to return to the Import Model From Rhapsody dialog.	
Rhapsody Files	List the file located in the associated project folder.	
Import	Click to import the Rhapsody model into Enterprise Architect.	
Close	Click to close this dialog.	
Help	Click to display this Help topic.	
Import Progress	Indicates the progress of the import.	

3.5.3.1.6 Limitations of XMI

Whilst XMI is a valuable means of specifying a UML model in a common format, it is relatively limited in the amount of additional information it can tolerate using the standard syntax.

Topics

Topic	Detail	See also
Notes on Enterprise Architect	<p>A lot of information from an Enterprise Architect model must be converted to Tagged Values, which import into other modeling systems as additional information or are ignored completely.</p> <p>Enterprise Architect can both generate and read:</p> <ul style="list-style-type: none"> • XMI 1.0 and 1.1 using UML 1.3 format • XMI 1.2 using UML 1.4 format, and • XMI 2.1 using UML 2.0 and UML 2.1 format <p>Round-tripping model elements using XMI in Enterprise Architect is possible using XMI 1.1 or XMI 2.1; XMI 1.1 uses the additional Tagged Values to store the UML 2.x information - this ensures model completeness when round-tripping UML 2.x model elements using XMI 1.1.</p> <p>Please note that Enterprise Architect exclusively uses XMI 1.1 for round-tripping models using Version Control.</p>	
Notes for Exporting to Rose and Other	There are discrepancies in the Unisys/Rose implementation with regard to spelling mistakes and slightly different syntax to the official	

Topic	Detail	See also
Tools	<p>XML 1.1 specification, so problems might occur.</p> <p>The way packages are arranged in different models can impact successful import into other systems; experimentation is the only work around for this problem.</p> <p>Some parts of the XML import/export process do not work as expected in products like Rational Rose; for example, Note Links are not supported, and State Operations import but do not appear in diagrams.</p> <p>Rational Rose only supports import of a full project, not a single package.</p> <p>For best results, it is recommended that you keep the model elements to export to Rose simple and conforming as closely as possible to the UML 1.3 specification.</p>	

3.5.3.1.7 The UML DTD

When you import or export Enterprise Architect packages to XML, the import or export process can be validated using a Data Type Definition (DTD).


Guide:

Topic	Detail	See also
Notes on DTDs	<p>The XML parser uses the DTD to validate the correctness of the model and to check that no syntactical errors have occurred.</p> <p>It is always best to use a DTD when moving packages between Enterprise Architect models as it ensures correctness of the XML output, and prevents attempted imports of incorrect XML.</p> <p>Several DTDs for XML/UML exist; the OMG defines a standard UML1.3 DTD for use in XML 1.1, and Enterprise Architect uses an extension of this with some additional element extensions for non-standard UML types, such as testing details.</p> <p>Whenever you read an XML file, the XML parser looks in the current directory for the DTD - if specified - using the DOCTYPE element in the XML file.</p> <p>If the parser cannot find the DTD, it records an error and aborts processing; you must ensure the UML_EA.DTD file is in the current XML output path (generated by default).</p>	

3.5.3.1.8 Controlled Packages

Controlled packages are a powerful means of 'externalizing' parts of an Enterprise Architect model.

Guide:

Topic	Detail	See also
Using Controlled Packages	<p>A controlled package is a package configured to save and load in XML format to a named file; the XML output is UML1.3 compliant XML, with Enterprise Architect extensions to support diagrams and additional model elements.</p> <p>Package XML is standard XML-compliant output that can be loaded into any XML viewer, or used by any XML-based tool to perform manipulations and extracts, such as document or code generators.</p> <p>Controlled packages appear in the Project Browser with a small colored rectangle to the left of the package icon, as shown below for the <i>CIM</i> package:</p>  <p>Using controlled packages you can support:</p> <ul style="list-style-type: none"> • Widely distributed development by having team members submit packages in the form of XML for import into a central Enterprise Architect repository • Version control, by writing model elements in XML text files suitable for version control using standard version control software <p>Using XML this way enables you to manually connect to third-party version control software outside the Enterprise Architect environment</p> <p>Enterprise Architect internally supports the configuration of version control through SCC and CVS</p> <ul style="list-style-type: none"> • Import and export of model elements between different models <p>For example, a Class library can be re-used in many models and kept up to date in target models using controlled packages, reloading packages as required when new versions of the Class model become available</p>	Version Control ^[312]

Notes

- If you are importing an XML 1.1 file that was previously exported with a UML_EA.DTD (Data Type Definition) file, the UML_EA.DTD file must be present in the directory into which the XML file is being written; an error occurs if the UML_EA.DTD file is absent
- When you select to apply a DTD during an XML 1.1 export, the UML_EA.DTD file is written to the output directory into which the XML files are written (unless the UML_EA.DTD file is already present in the directory); no error is generated if the UML_EA.DTD file is not present in this directory during the XML export

3.5.3.1.8.1 Controlled Package Menu

Access **Project Browser > package | Package Control**

This menu is for a package that is not itself under version control (but that might contain child packages that are under version control).

Action	Usage	Shortcut	See also
Configure (various settings for the package)	Display the Package Control Options dialog, which enables you to specify whether this package (and its child packages) is controlled and which file it is controlled through.	Ctrl + Alt + P	Configure Packages ^[426]
Manage Baselines	Create a Baseline of the current package, or compare the current package with a previous Baseline.	Ctrl + Alt + B	Managing Baselines ^[398]
Check In Branch	For the selected branch of the model, (that is, the selected package and all of its child packages) display the Select Packages to Check In dialog, listing all version controlled packages within that branch that are checked out to you. You can then select packages in the displayed list, to be submitted for check-in.		Check in a Model Branch ^[364]
Check Out Branch	For the selected package, check out the package and recursively check out all of its contained sub-packages. Enterprise Architect retrieves the latest version of the packages from the central repository, overwriting the current packages. After check out, the packages are available for editing.		Check Out a Model Branch ^[365]
Save package to file	Save a controlled package to an XML file.	Ctrl + Alt + S	Save a Package ^[428]
Load package from file	Load a previously-saved XML file.	Ctrl + Alt + L	Load a Package ^[429]

Action	Usage	Shortcut	See also
View package XMI	Display the package XMI after the package has been exported to XMI.	Ctrl + Alt + X	
Compare with XMI file	(Package not under version control) Compare the current package with a previously-saved XMI file of the package.		The Compare Utility (Diff) ^[40]
Add Branch to Version Control	Apply version control to all packages within a selected model branch, in a single operation.		Apply Version Control to Branches ^[37]
Export as Model Branch	Export a newly created model branch from your own private copy of a model.		Export Controlled Model Branch ^[37]
Import a Model Branch	Retrieve a model branch and import it into either the source model or another model.		Import Controlled Model Branch ^[37]
Get package (for version control)	Enables you to gain access from packages in the version-controlled repository that is currently available in your model.		Version Control ^[31]
Get All Latest (for version control)	Retrieve the latest version of the package from the repository. Available only for packages that are checked in. The alternative option Get Latest - if displayed - is not intended for sharing .EAP files and should only be used when users have their own individual databases.		Version Control ^[31]
Re-synch Status With VC Provider	(Version controlled package.) Update the version control status value recorded for the selected package in the Enterprise Architect project to match the value reported by the version control provider, without performing an XMI import or export.		
Version Control Settings	Display the Version Control Settings dialog.		Version Control Settings Dialog ^[34]
Update Package Status	Provide a bulk update on the status of a package. This includes status options such as Proposed , Validate and Mandatory .		Update Package Status ^[52]

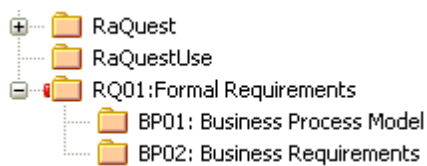
Learn more

- For a package that is directly under version control, see [Package Version Control Menu](#)^[357]

3.5.3.1.8.2 Configure Packages

Before you can use a controlled package, you must configure it with options such as the filename to save to/load from, the type of export and the version number.

Once a package is configured and marked as controlled, it is displayed in the Project Browser with a small colored rectangle next to the package icon, indicating it is a controlled package; for example, the *RQ01: Formal Requirements* package below.



Access [Project Browser](#) | [Package Context Menu](#) | [Package Control](#) | [Configure](#)

How to

To configure a controlled package

Step	Action	See also
1	Select the package to control or configure, and the Configure context menu option. The Package Control Options dialog displays.	
2	Select the Control Package checkbox to indicate that this is a controlled package.	
3	Click on the Version Control drop-down arrow and select the version control repositories; this connects the package to a specific version control configuration.	
4	In the XMI Filename field, type or browse for the path and XML file for importing and exporting XML files. The field accepts <i>Local Path Substitution</i> strings; for example, use an XML local path definition where: <code>myLocal Path=" C: \ Documents and Settings\ John\ Desk top\ EA Model s "</code> Then %myLocal Pat h% Cl M. xml is equivalent to: <code>C: \ Documents and Settings\ John\ Desk top\ EA Model s \ Cl M. xml</code>	
5	In the UML/XMI Type field, click on the drop-down arrow and select the type of XML generated; options include:	

Step	Action	See also
	<ul style="list-style-type: none"> • Enterprise Architect XMI/UML 1.3 • Rational Rose/Unisys UML 1.3 and • Generic XMI 1.0/UML 1.3 <p>Currently only Enterprise Architect UML 1.3 is supported for complete import/export round tripping of packages.</p>	
6	In the Version ID field, type the version ID number.	
7	In the Owner field, type or select the name of the package owner.	
8	If required, click on the Use DTD checkbox to use a Data Type Definition (DTD).	
9	If required, click on the Log Import/Export checkbox to log import and export activity to a log file.	
10	If required, click on the Batch Import checkbox to mark the package as a Batch Import package.	
11	If required, click on the Batch Export checkbox to mark the package as a Batch Export package.	
12	<p>If required, click on the Include sub-package checkbox to deselect it, to include only the immediate contents of the package in an XML export (XML stubs); this is available only for an XML 1.1/UML 1.3 export.</p> <p>If you leave the checkbox selected, the entire sub-package hierarchy of this branch is included in the export.</p>	
13	Click on the OK button to set the Package Control options.	

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Configure Packages** permission to configure controlled packages and package properties
- For batch import, the file date of the XML file is stored; you can bypass the batch import if there is no change; that is, the file date of the last import matches that of the current file

Learn more

- [List of Available Permissions](#)^[264]

3.5.3.1.8.3 Remove Package from Control

If required, you can remove the control from a package. Before removing the package control, you must check in the package if it is being used for version control.

Access [Project Browser](#) | [Package Context Menu](#) | [Package Control](#) | [Configure](#) (Ctrl+Alt+P)

How to

To remove control from a package

Step	Action	See also
1	Select the package and the Configure menu option. The Package Control Options dialog displays.	
2	Deselect the Control Package checkbox. If the package is under version control, this sets the Version Control field to (None) .	
3	Click on the OK button to remove package control. Package control for the selected package has now been removed.	

Notes

- When disconnecting a package from version control, the association between the package and the exported XML file is removed from your model; however, the XML file itself is not removed from version control, nor is it deleted from your local version control working copy folder

This is because it is possible for another model to be using the version controlled package and still referencing the associated version controlled XML file

Learn more

- [Check In a Package](#)^[365]

3.5.3.1.8.4 Save a Package

Once you have correctly configured a package, you can save the controlled package to the designated XML file.

Access [Project Browser](#) | [Package Context Menu](#) | [Package Control](#) | [Save Package to File](#)

How to

To save a controlled package to file

Step	Action	See also
1	Select the package and the Save package to File menu option. The export process executes automatically according to your configured preferences, overwriting any existing file.	

Notes

- If you are using a version control package in conjunction with the exported package files, you must check out the XMI file first to enable Enterprise Architect to overwrite the existing version

Learn more

- [Configure Packages](#) ⁴²⁶

3.5.3.1.8.5 Load a Package

If a package has been marked for control it is displayed in the Project Browser with a red rectangle to the left of the package icon. If you have previously saved the controlled package, you can reload it.

Access [Project Browser](#) | [Package Context Menu](#) | [Package Control](#) | [Load package from file](#)

How to

To load a controlled package from an XMI file

Step	Action	See also
1	Select the package and the Load package from file menu option. If you have configured the package control details, Enterprise Architect prompts you to confirm the import.	
2	Click on the Yes button to confirm the import. The current package is deleted and the saved package is imported.	

Notes

- Importing deletes the current package entirely from the model, and the action cannot be undone; you must be careful not to lose any current changes or information

Learn more

- [Configure Packages](#)^[426]

3.5.3.1.8.6 Batch XMI Export

You can export a group of controlled packages to XMI in one step.

Access [Project | Model Import/Export | Batch XMI Export](#)

How to

To export a group of controlled packages to XMI

Step	Action	See also
1	Select the Batch XMI Export menu option. The Batch XMI Export dialog displays.	
2	Select the checkbox against each package to include in this export run. Select the Select All checkbox to select all packages in the list.	
3	Select the Restore Cross References Without Prompting checkbox to automatically restore any missing cross package references without displaying a prompt.	Report Deletion of Cross- Package References ^[433]
3	To save this configuration as the default, click on the Save Settings button.	
4	Click on the Run Export button. Enterprise Architect cycles through each checked package and exports it using the options specified in the Package Control Options dialog. As long as a valid filename exists, Enterprise Architect exports the package to XMI and proceeds to the next package.	Configure Packages ^[426]

Notes

- The **Restore Cross References Without Prompting** checkbox is enabled only if the option **Report Cross Package Reference Deletions (XMI 1.1)** is set on the XML Specifications page of the Options dialog
- The option is applicable only to XMI 1.1 controlled packages

Learn more

- [XML Specifications](#) ⁵⁶²

3.5.3.1.8.7 Batch XMI Import

You can import a group of controlled packages from XMI into Enterprise Architect in one step.

As Enterprise Architect processes each package, it updates the **Status** column against each package name on the Batch XMI Import dialog.

- If the import is successful, Enterprise Architect updates the package status to **Imported**
- If the import is unsuccessful, Enterprise Architect updates the package status to **Not Imported**

Common reasons for an import to fail include:

- The package not being correctly configured
- The last import file date matches the import date of the file currently on disk

Access [Project](#) | [Model Import/Export](#) | [Batch XMI Import](#)

How to

To import a group of controlled packages from XMI

Step	Action	See also
1	Select the Batch XMI Import menu option. The Batch XMI Import dialog displays.	
2	Select the checkbox against each package to include in the import. Select the Select All checkbox to select all packages in the list.	
3	To save this configuration as the default, click on the Save Settings button.	
4	Click on the Run Import button. Enterprise Architect cycles through each checked package and imports it.	Configure Packages ⁴²⁶

Notes

- To avoid re-importing the same module multiple times, select the **Check file date before import** checkbox; Enterprise Architect then does not import a file if the last import file date matches that of the one currently on disk

3.5.3.1.8.8 Manual Version Control with XMI

You can use XMI to support version control by writing model elements in XML text files suitable for use with standard version control software.

Using XMI in this manner enables you to manually connect to third-party version control software outside the Enterprise Architect environment.

Enterprise Architect internally supports the configuration of version control through SCC and CVS configurations.

How to

Before using XMI for version control

Step	Action	See also
1	Select suitable packages in the Project Browser, to be marked as controlled packages.	
2	Configure these with filenames that are visible to a version control system of your choice.	Configure Packages ^[426]
3	Save the controlled packages to establish a model base and check these into the version control system.	Save a Package ^[428]

To apply version control

Step	Action	See also
1	Continue working on a package until versioning is required.	
2	Check out the package XMI file from the version control system.	
3	Save the relevant package using the controlled package support.	
4	Check the package back into the version control system.	

To recover an earlier version

Step	Action	See also
1	Save the current version first. This is important, because the package is completely deleted during the import process. If necessary, manually update the version control system.	
2	Get the required package version from the version control system.	
3	Select the package to reload.	
4	Select the Package Control Load package from file menu option to import the previous version. Enterprise Architect deletes the controlled package and restores the previous version.	Load a Package [429]

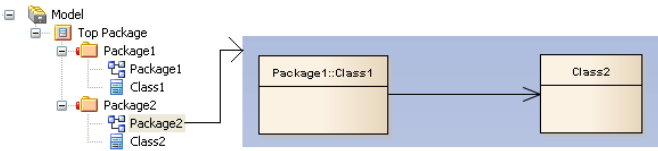
Learn more

- [Version Control](#) [312]

3.5.3.1.8.9 Report Deletion of Cross Package References

When saving an XMI 1.1 controlled package, if the **Report Cross Package Reference Deletions (XMI 1.1)** option is set in the Options dialog, Enterprise Architect compares the package being exported with the latest XMI 1.1 revision of the package.

Topics

Topic	Detail	See also
Scenario	<p><i>Master.eap</i> contains two controlled packages - <i>Package1</i> and <i>Package2</i>.</p>  <p>You now import just the controlled package, <i>Package2</i>, into a new <i>.eap</i> file, <i>New.eap</i>.</p> <p>Since <i>New.eap</i> does not contain <i>Package1</i> (and hence its element <i>Class1</i>), Enterprise Architect cannot show <i>Class1</i> and the Association connector on the diagram <i>Package2</i>.</p> <p>When the package <i>Package2</i> is exported to XMI from <i>New</i>.</p>	

Topic	Detail	See also
	<p><i>eap</i>, the XML file does not contain the information that the diagram <i>Package2</i> used to contain the element <i>Class1</i> and the Association connector.</p> <p>If you import this XML file into <i>Master.eap</i>, Enterprise Architect removes the element <i>Class1</i> and the Association connector from the diagram <i>Package2</i>.</p>	
Identify Cross-Package References	<p>In order to retain such cross-package references in XML 1.1, you can set the Report Cross Package Reference Deletions (XML 1.1) option so that when you save a controlled package, Enterprise Architect compares the latest XML 1.1 revision of the package with the package being exported.</p> <p>Any items that are missing from the package are listed on the Missing Cross References dialog.</p>	XML Specifications [562]
Actions	<p>To merge missing items, as identified in the Missing Cross References dialog, into the XML file that is being saved:</p> <ul style="list-style-type: none"> • Select the checkbox against each item type • Click on the OK button <p>If you do not select any items, they are omitted from the saved XML file.</p> <p>To abort or cancel this XML Export (or Check-in, or Put Latest) operation:</p> <ul style="list-style-type: none"> • Click on the Cancel button <p>To review the details of an entry:</p> <ul style="list-style-type: none"> • Double-click on the type <p>The details of each item are shown on the Missing <item type> dialog.</p> <p>You can copy a value from any of the columns in this dialog to the clipboard:</p> <ul style="list-style-type: none"> • Click on the column value to highlight it; the selected value is shown in the bottom left corner of the dialog • Copy that value to the clipboard (Ctrl+C) 	

3.5.3.2 CSV Import and Export

Enterprise Architect enables you to import and export information about Enterprise Architect elements to and from CSV text files.

Topics

Topic	Detail	See also
Specifications	To import and export element data from Enterprise Architect using CSV files, you must first set up one or more file specifications. You do this because the specification defines what types of value from the spreadsheet are to be imported, and how the information is translated between the spreadsheet and Enterprise Architect.	CSV Specifications ^[435]
Import From CSV	Once you have defined a CSV import specification, you can read in major element characteristics from a CSV text file.	CSV Import ^[440]
Export To CSV	Once you have defined a CSV export specification it is possible to write out major element characteristics to a CSV text file.	CSV Export ^[439]

3.5.3.2.1 CSV Specifications

To import and export element data to and from Enterprise Architect using CSV files, you must first set up one or more file specifications.

A file specification defines the:

- Fields from the spreadsheet in the order they are imported or exported
- Filename (optional) and
- Delimiter between columns

Once you have defined one or more specifications, one can be selected in the CSV Import/Export dialog as the current specification to apply during an import or export action.

CSV Import/Exports only imports and exports *elements* (within packages) and their properties; items such as Class attributes cannot be imported or exported through this mechanism.

XML import/export provides a solution to this limitation, as does use of the Automation Interface (Object Model).

Access [Project | Model Import/Export | CSV Import/Export Specifications](#)

Reference

On the CSV Import/Export Specification dialog, complete the fields as required, as indicated in the following table:

Field	Usage	See also
Specification Name	Select the unique name for this specification.	
Delimiter	Specify the character delimiter to use between record fields. If a field contains an instance of the delimiter, the field is exported wrapped in " (quotation marks) and all instances of " in the field are doubled (that is, " becomes "").	
Notes	Record a brief description of the specification.	
Default Filename	Select the default filename.	
Default Direction	Set the default action - Import or Export . A specification can be used in either direction, but this enables you to set the default type.	
Default Types	Limit the element types being exported, by entering a comma-separated list: for example: <i>class,requirement,component,node,object</i> . <ul style="list-style-type: none"> • If you specify element types, ONLY elements of those types are exported or imported; therefore, in order to enable the Preserve Hierarchy option to operate (if selected) you must include Package as an element type, otherwise there are no packages in which to preserve the hierarchy • If you do not specify any default element types, all elements including Packages are exported or imported and the hierarchy can be preserved 	
Preserve Hierarchy	Include fields generated by Enterprise Architect to embed/reconstruct the package hierarchy.	Using Preserve Hierarchy ^[438]
Available Fields	Select from a list of possible record fields, not yet allocated.	
File Specification	List the record fields (in the order they are plotted across the spreadsheet) already assigned.	
Add Tagged Value Field	Adds a Tagged Value column to the CSV Specification, to import or export the value that an element has for a	

Field	Usage	See also
	<p>particular tag.</p> <p>When you click on this button, a prompt displays to select one of two options:</p> <ul style="list-style-type: none"> • Value - to import or export the content of the Tagged Value Value field; a column heading with the prefix <i>TagValue_</i> is added to the specification • Notes - to import or export the content of the Tagged Value Notes field (some Tagged Values use the Notes field to accommodate long values, such as <memo> text); a column heading with the prefix <i>TagNotes_</i> is added to the specification <p>In each case, the Add Tag Value Column dialog displays. Select either the</p> <ul style="list-style-type: none"> • Global Tagged Values option, or • Tagged Values from the selected Element option, and select the required element using the Other Element button <p>Now select the required Tagged Value from the Tagged Value drop-down list, and click on the OK button.</p>	Package Browser Options ⁵⁹⁵
Add Field	Move all selected fields in the top list to the bottom list.	
Remove Field	Move all selected fields in the bottom list back to the available list.	
New	Create a new specification.	
Save	Save changes to the currently selected specification.	
Save As	Save the current specification with a new name.	
Delete	Delete the current specification.	
Close	Close this dialog.	

Notes

- In **Available Fields** and **File Specification**, the record fields **Created Date** and **Modified Date** are not set when imported from CSV

Learn more

- [XML Import and Export](#)^[41]
- [Enterprise Architect Object Model](#)^[238]

3.5.3.2.1.1 Using Preserve Hierarchy

The **Preserve Hierarchy** option on the CSV Import/Export Specification dialog includes fields generated by Enterprise Architect to embed/reconstruct the package hierarchy in a CSV import or export.

When selected, the **Preserve Hierarchy** option inserts two fields into the CSV specification that are:

- automatically populated by Enterprise Architect on export and
- used to reconstruct the exported package's hierarchy upon import

Field	Usage	See also
CSV_KEY	A unique identifier for the current element. This key is unique per export; subsequent exports produce different keys for the same set of elements.	
CSV_PARENT_KEY	The corresponding CSV_KEY of the current element's parent. If the field is left blank or references a non-existent CSV_KEY, the element is added to the top level of the package.	

It is highly recommended that you do not change these fields if they have been automatically generated by Enterprise Architect's CSV exporter.

If you intend to import hierarchical information from a spreadsheet that was **not** populated by exporting data from Enterprise Architect, you must add these two fields to your spreadsheet as the last two columns, and populate the columns yourself.

For example:

NAME	TYPE	NOTES	PRIORITY	STATUS	CSV_KEY	CSV_PARENT_KEY
Requirement Package	Package	Notes Package1			Package1	
REQ1	Requirement	Notes on REQ1	High	Approved	REQ1	Package1
REQ2	Requirement	Notes on REQ2	High	Approved	REQ2	Package1
REQ2.1	Requirement	Notes on REQ2.1	High	Approved	REQ2.1	REQ2
REQ2.2	Requirement	Notes on REQ2.2	Med	Approved	REQ2.2	REQ2
REQ2.3	Requirement	Notes on REQ2.3	High	Approved	REQ2.3	REQ2
REQ3	Requirement	Notes on REQ3	High	Approved	REQ3	Package1
REQ3.1	Requirement	Notes on REQ3.1	High	Approved	REQ3.1	REQ3
REQ3.2	Requirement	Notes on REQ3.2	High	Approved	REQ3.2	REQ3
REQ4	Requirement	Notes on REQ4	High	Approved	REQ4	Package1
REQ4.1	Requirement	Notes on REQ4.1	High	Approved	REQ4.1	REQ4
REQ4.2	Requirement	Notes on REQ4.2	High	Approved	REQ4.2	REQ4
REQ4.3	Requirement	Notes on REQ4.3	High	Approved	REQ4.3	REQ4

3.5.3.2.2 CSV Export

Enterprise Architect enables you to export information about elements to a CSV text file, using a CSV export specification.

If you intend to *re-import* the exported information into Enterprise Architect at some point, it is recommended that you include the **GUID** field in the CSV export specification. This ensures that Enterprise Architect can identify and update existing elements, rather than creating duplicates.

Access [Project Browser](#) | [Package Context Menu](#) | [Import/Export](#) | [CSV Import/Export](#)

How to

To export data in CSV format, follow the steps below

1. In the Project Browser, right-click on the package containing the elements to export and select the **CSV Import/Export** menu option.
2. The CSV Import/Export dialog displays; complete the fields as required, as indicated in the following table:

Field	Usage	See also
Package	Confirm the name of the current selected package.	
Specification	Specify the name of the export specification ^[413] to use.	CSV Specifications ^[435]
Edit/New	Edit the export specification or create a new one.	
File	Specify the filename to export to.	
Types	List the element types to export: leave blank for all, or enter a comma-separated list of types.	

Field	Usage	See also
	<p>If you specify element types, ONLY elements of those types are exported; therefore, to enable the Preserve Hierarchy option in the specification to operate (if selected) you must include <i>Package</i> as an element type, otherwise no Packages are exported in which to preserve the hierarchy.</p> <p>If you do not specify any element types, all elements including Packages are exported and the hierarchy can be preserved.</p>	
Action	Select the Export radio button to export to file.	
Print Results	Print out the result list.	
View File	View the resultant CSV file with the default Windows application for CSV files.	
Run	Perform the export.	
Close	Exit this dialog.	

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have both **Export XMI** and **Import XMI** permissions to use the **CSV Import/Export** option

Learn more

- [List of Available Permissions](#) ²⁶⁴

3.5.3.2.3 CSV Import

Enterprise Architect enables you to import information about elements from a CSV text file, using a CSV export specification.

You import the CSV file into a selected package; if this package or any element within the package has a lock on it, you cannot import the CSV file into it. The **Import** option on the dialog is grayed out.

The format and content of the source data file should resemble the following:

The screenshot shows a spreadsheet with the following data:

	A	B	C	D	E	F	G	H
1	Name	Type	Notes	Phase	Version	Status	Priority	Difficulty
2	FR27 – Retrieval of histo	Requirement	The system must be able t	1	1	Approved	Medium	Medium
3	FR28 – Transmission of	Requirement		1	1	Approved	Medium	Medium
4	FR29 – The staff must n	Requirement	The system must use stan	1	1	Mandatory	Low	Medium
5	FR30 – The Interface mu	Requirement	Critical to the success of t	1	1	Approved	High	Medium
6	FR44 – The interface sho	Requirement	An important aspect of the	1	1	Approved	High	Medium
7	FR5 – 2000 hours mean	Requirement	The mean time between fai	1	1	Proposed	Medium	Medium
8	FR6 – Must be recovera	Requirement	In the event of software or l	1	1	Approved	Medium	High
9	FR7 – 99.999% accurac	Requirement	The system accuracy defin	1	1	Approved	Medium	Medium

Access [Project Browser](#) | [Package Context Menu](#) | [Import/Export](#) | [CSV Import/Export](#)

How to

To import data in CSV format

1. In the Project Browser, right-click on the package to import into and select the **CSV Import/Export** context menu option.
2. The CSV Import/Export dialog displays; set the required options; as outlined below:

Field	Usage	See also
Package	Confirm the name of the current selected package.	
Specification	Specify the name of the import specification [2226] to use.	Import specification [435]
Edit/New	Edit the import specification or create a new one.	
File	Specify the spreadsheet filename to import from.	
Types	Not used for import.	
Action	Select the Import radio button to import from the file (grayed-out if the package or a child item in the package is locked).	
Print Results	Print out the result list.	
View File	View the source CSV file with the default Windows application for CSV files.	
Run	Perform the import.	

Field	Usage	See also
Close	Exit this dialog.	

Notes

- When importing, Enterprise Architect checks the specification to see if there is a GUID field included; if there is, Enterprise Architect attempts to locate the element identified by the GUID and, if successful, updates the current element rather than creating a new one
- If no GUID field is defined, or Enterprise Architect cannot locate the identified element, a new element is created and placed in the current package; during import, **Type** is a mandatory field in the source file and must match one or more of the legal Enterprise Architect element types: for example: Requirement, or Class
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have both **Export XMI** and **Import XMI** permissions to use the **CSV Import/Export** option

Learn more

- [List of Available Permissions](#)^[264]
- [Enterprise Architect element types](#)^[2487]

3.5.3.3 Perform a Project Data Transfer

The Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect support a range of data repositories.

At some point, it might become necessary to move a complete model from one repository to another, row by row, table by table.


Access **Tools | Data Management | Project Transfer**

Use to

- Upload an existing .EAP file to a SQL Server or MySQL repository
- Download a repository in MySQL or SQL Server to a .EAP file
- Move a repository from SQL Server to MySQL or from one server to another
- Move all records from a .EAP file with replication to a model with none (Remove Replication)
- Copy all records from a .EAP file to another (recommended after serious network crash or repeated database corruption)
- Copy all records from a JET 3.5 to JET 4 (Access 2000 or XP) repository - or back the other way

How to

To perform a project data transfer

Step	Action	See also
1	Take a backup of the target project to ensure that you can recover any important information it contains.	
2	Select the Project Transfer menu option. The Project Transfer dialog displays.	
3	Click on the option for the required transfer type: <ul style="list-style-type: none"> • .EAP to .EAP • DBMS to .EAP • .EAP to DBMS • DBMS to DBMS 	
4	In the Source Project and Target Project fields, type or select the name or connection string for the Source and Target projects.	
5	If you want to capture the transfer in a log file, select the Logfile checkbox and browse for the appropriate log file location.	
6	Click on the Transfer button.	
7	It is good practice to do a Project Compare after this process to verify that all records are written.	Compare Projects 

Notes

- All records in the target repository are overwritten
- You cannot move a model from a source .EAP file of a version earlier than 3.5.0
- You must transfer data into a repository (where scripts have been run to set up tables), not just a database; if necessary
 - Install the DBMS software and create a database; ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic
 - Run a script supplied by Sparx Systems to create the required tables
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Transfer Data** permission to transfer project data between repositories

Learn more

- [Sparx Systems Scripts webpage](#)
- [Create a Repository](#)^[194]
- [List of Available Permissions](#)^[264]

3.5.3.3.1 Copy Packages Between Projects

Using the XML import/export capabilities of Enterprise Architect, you can copy and move packages between Enterprise Architect projects.

This gives you a high level of flexibility in building a project from re-usable parts and from elements produced in widely-dispersed geographic regions.

Access [Project Browser | Package Context Menu | Import/Export | Export package to XMI file](#)
[Project Browser | Package Context Menu | Import/Export | Import package from XMI file](#)

Use to

- Copy individual child packages within or between models
- Duplicate larger structures, such as complete models or projects

How to

To copy a package from one Enterprise Architect project to another

Step	Action	See also
1	Open the Enterprise Architect project to copy <i>from</i> .	
2	In the Project Browser, right-click on the package to copy and select the Export package to XMI file option. The Export Package to XMI dialog displays.	
3	Select the appropriate options and filename.	Export to XMI ^[413]
4	Click on the Export button to begin the export process.	
5	When the export is complete, open the Enterprise Architect project to copy <i>into</i> .	
6	In the Project Browser, navigate to the location to import the package into.	
7	Right-click and select the Import package from XMI file option.	

Step	Action	See also
	The Import Package from XML dialog displays.	
8	Select the appropriate options and filename.	Import from XML ^[417]
9	Click on the Import button. The package is copied from the source project to the destination project.	

Notes

- If the package you are importing already exists in the target project (that is, it has been imported previously), you must either import *over* the existing package or select the **Strip GUIDs** option

In the latter case, Enterprise Architect creates a copy of the original package

Learn more

- [Copy a Package](#)^[683]
- [Copy Elements Between Packages](#)^[808]

3.5.4 Compare Projects

It is often useful to compare the size and row counts of two projects; for example, after:

- A database crash
- Importing from XML, or
- Deleting model elements

Enterprise Architect examines the number of rows in each database and produces a report indicating the total records in each and the difference in record count between the two. No examination is made of the data in the table.

Comparing projects this way is a convenient 'sanity check' after restoring a backup or doing a project data transfer. If discrepancies are found, you must investigate further manually.

Access [Tools | Data Management | Project Compare](#)

Use to

- Compare a .EAP file to another .EAP file
- Compare a .EAP file to a DBMS-based repository
- Compare two DBMS repositories

How to

Step	Action	See also
1	Select the Project Compare menu option. The Project Compare dialog displays.	
2	Select the required comparison: <ul style="list-style-type: none">• .EAP to .EAP• DBMS to .EAP• .EAP to DBMS• DBMS to DBMS	
3	In the Source Project and Target Project fields, type the name or connection string for the Source and Target projects.	
4	Click on the Compare Projects button. The results of the comparison display in the panel at the bottom of the dialog.	
5	If required, click on the Print List button to print the results.	

3.6 Project Management



Enterprise Architect provides strong support for:

- Project Management, in estimating project size, measuring risk and effort, and assigning resources to elements
- Change control and maintenance

Topics

Topic	Detail	See also
Metrics and Estimation	<p>Project estimation is working out how much time and effort is required to build and deploy a solution.</p> <p>Enterprise Architect provides the Use Case metrics facility as a means of:</p> <ul style="list-style-type: none"> • Measuring the complexity of a system • Getting an indication of the effort required to implement the model • Getting an indication of the project timescale <p>You base these estimates on carefully-calibrated metrics.</p>	Use Case Estimation ^[516]
Resource Management	<p>Resources are the people who work on a project.</p> <p>You can assign roles to resources and allocate tasks on specific model elements, which enables tracking of effort and estimation of time to complete.</p>	Project Resources ^[449]
Project Maintenance	<p>During a project you monitor and manage the development and progress of individual model elements.</p> <p>You can record problems, changes, issues and tasks that affect these individual elements as they arise, and document the solution and associated details.</p> <p>Similarly, Enterprise Architect helps you to manage changes and issues that apply to the whole system.</p>	Maintenance ^[2236] Changes and Defects ^[2244]
Project Tasks and Issues	<p>In the course of a project, there are various non-technical tasks that are vital to the successful management and completion of the project, such as meetings.</p> <p>Enterprise Architect helps you to record and monitor these, and to manage non-technical project issues as</p>	Project Tasks ^[460] Project Issues ^[462] Project Task Allocation ^[472]

Topic	Detail	See also
	they arise.	

Learn more

- [The Project Management Window](#)^[448]
- [Personal Information](#)^[486]
- [The Project Information Window](#)^[460]
- [Project Calendar](#)^[502]
- [Project Glossary](#)^[467]
- [Update Package Status](#)^[529]
- [Manage Bookmarks](#)^[530]
- [Monitor Change Events](#)^[532]

3.6.1 The Project Management Window

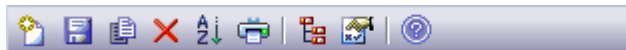
The Project Management window enables you to input the following quantities associated with an element contained in the model:

- Resources - the people who work on a project, who can be assigned roles and allocated tasks
- Effort - effort expended in work on the element
- Risks - risk associated with the element
- Metrics - metrics measured for an element

Each of these four quantities has a separate tab in the window.

Access [View | More Element Tools | Project Management](#)

Reference



Action	Usage	See also
New	Create a new item.	Resource Allocation ^[450] Effort Management ^[452] Risk Management ^[453] Metrics ^[455]
Save	Save changes to an item.	

Save As New	Duplicate an existing entry. You must change an item's Role for this to become enabled .	
Delete	Delete an item from the list.	
Sort	Sort the items in the list into alphabetical order.	
Print	Print item data from the list.	
Browse Element	Display the Element Browser window for the selected element, to list and select the project management items for the element.	The Element Browser 18761
Show/Hide Properties	Swap between detailed and summary window styles.	
Help	Show help contents for this window.	

Notes

- Click on an element in the Project Browser to display project management information for that element in the Project Management window
- The window has two formats:
 - *List mode* - provides a detailed list showing several columns of information
 - *Item mode* - a summary list plus an editor to add or edit information for a single selected item
- Toggle between these modes using the **Show/Hide Properties** button in the window toolbar; each tab toggles between Item mode and List mode independently
- An asterisk on a tab name indicates that the tab contains saved information; if there is no information for a category of item, or the information has not yet been saved, its tab has no asterisk
- Right-click on the list to view the context menu, which enables you to add and delete list items

3.6.2 Project Resources

Resources are the people who work on a project. They can be assigned roles and allocated tasks, which enables tracking of effort and estimation of time to complete.

Resources are added, modified and deleted using the Project Management window.

Access [View | More Element Tools | Project Management](#) (Ctrl+Shift+7)

Use to

Usage	See also
Allocate a resource to an element.	Resource Allocation ^[450]
Record additional project management information for an element.	Effort Management ^[452] Risk Management ^[453] Metrics ^[455]
Obtain a report of resource allocation details.	Resource Report ^[477]
Configure Project Management data and populate the drop-down lists used on the Project Management dialog tabs.	Roles ^[991] Clients ^[993] Effort Types ^[456] Metric Types ^[457] Risk Types ^[458]

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Project Information** permission to update and manage project resources, effort, metrics and risks

Learn more

- [The Project Management Window](#) ^[448]
- [List of Available Permissions](#) ^[264]

3.6.2.1 Resource Allocation

In Enterprise Architect, the Project Manager can connect a named resource in a named role to a given model element.

To select the element to which to allocate resources, click on the required element in the Project Browser.

Access [View](#) | [More Element Tools](#) | [Project Management](#) > [Resource Allocation: New](#) (Ctrl+Shift+7)

Use to

- Track how far development of required components and Classes has progressed

How to

To enter resource allocation details for an element

Step	Action	See also
1	Select the element in the Project Browser.	
2	Select the Project Management menu option. The Project Management window displays, showing the Resource Allocation tab.	The Project Management Window [448]
3	Click on the New icon on the Project Management window toolbar.	
4	On the Resource Allocation tab enter the: <ul style="list-style-type: none"> Name of the resource (click on the drop-down arrow and select, or type the name in) Role of the resource (click on the drop-down arrow and select, or type the name in) Start and end date for the availability of the resource Time allocated to the resource Percentage of the task the resource has completed Expected time allocated to the resource Actual time expended by the resource Description of the work being done by the resource (this text is also displayed in the Notes window; it cannot be edited in that window) Notes on the activity history of the work done by the resource (this text is also displayed in the Notes window; it cannot be edited in that window) 	Notes [977]
5	Click on the Save icon in the Project Management window toolbar.	

Notes

- To edit existing Resource Allocation items for this element, click on the required item in the:
 - List panel to the left of the window, in *Item* mode
 - List, in *List* mode
 - Project Management folder in the Element Browser window - if this window is not displayed, click on the **Browse Element** icon in the Project Management window toolbar; resource allocation item icons have an **R** in the bottom right corner

Learn more

- [The Element Browser](#)^[876]

3.6.2.2 Effort Management

In Enterprise Architect, the Project Manager can allocate effort (as time) to work on a given model element.

To select the element to which to allocate effort, click on the required element in the Project Browser.

Access **View | More Element Tools | Project Management > Effort: New** (Ctrl+Shift+7)

Use to

- Enter the effort allocated to an element

How to

To enter effort allocation details for an element

Step	Action	See also
1	Select the element in the Project Browser.	
2	Select the Project Management menu option. The Project Management window displays, showing the Resource Allocation tab.	
3	Click on the Effort tab.	
4	Click on the New icon on the Project Management window toolbar.	
5	On the Effort tab enter the: <ul style="list-style-type: none"> • Name of the effort (a short description) • Type of the effort (type the name or click on the drop-down arrow and select; the selection list is drawn from the global Effort Type list, but any new efforts you type in the Type field are not added to the list) • Time the effort is expected to expend • Notes on effort (this text is also displayed in the Notes window; it cannot be edited in that window) 	Effort Types ^[456] Notes ^[977]
6	Click on the Save icon in the Project Management window toolbar.	

Notes

- To edit existing Effort items for this element, click on the required item in the:
 - List panel to the left of the window, in *Item* mode
 - List, in List mode, or
 - Project Management folder in the Element Browser window - if this window is not displayed, click on the **Browse Element** icon in the Project Management window toolbar; effort item icons have an **E** in the bottom right corner
- Although Enterprise Architect does not currently provide detailed reports on effort within a model, you can use the Automation Interface or similar tools to create your own custom reports based on effort information you enter

Learn more

- [The Element Browser](#)⁸⁷⁶
- [Enterprise Architect Object Model](#)²³⁸⁷

3.6.2.3 Risk Management

In Enterprise Architect, the Project Manager can allocate the possible weighting of defined risks to work on a given model element.

To select the element to which to allocate risk weightings, click on the required element in the Project Browser.

Access **View | More Element Tools | Project Management > Risks: New (Ctrl+Shift+7)**

Use to

- Enter the details of risks that might impact work on an element

How to

To enter risk details for an element

Step	Action	See also
1	Select the element in the Project Browser.	
2	Select the Project Management menu option. The Project Management window displays, showing the Resource Allocation tab.	
3	Click on the Risks tab.	

Step	Action	See also
4	Click on the New icon on the Project Management window toolbar.	
5	On the Risks tab enter the: <ul style="list-style-type: none"> Name of the risk (a short description) Type of the risk (Mandatory: type the name or click on the drop-down arrow and select; the selection list is drawn from the global Risk Type list, but any new risks you type in the Type field are not added to the list) Weighting allocated to the risk Notes on the risk (this text is also displayed in the Notes window; it cannot be edited in that window) 	Risk Types ^[458] Notes ^[977]
6	Click on the Save icon in the Project Management window toolbar.	

Notes

- The risks described here are not the same as those represented by Risk elements; the risks described above are properties of a single element, whilst the Risk element represents something that can impact a range of other elements
- If you select a global risk type from the **Type** drop-down list and the associated **Weight** field is empty, the default Weight value is allocated to the **Weight** field on the Risks tab
- To edit existing Risk items for this element, click on the required item in the:
 - List panel to the left of the window, in *Item* mode
 - List, in List mode
 - Project Management folder in the Element Browser window - if this window is not displayed, click on the **Browse Element** icon in the Project Management window toolbar; risk item icons have an **Ri** in the bottom right corner
- Although Enterprise Architect does not currently provide detailed reports on risks within a model, you can use the Automation Interface or similar tools to create your own custom reports based on risk information you enter

Learn more

- [The Element Browser](#)^[876]
- [Enterprise Architect Object Model](#)^[2381]
- [Risk Elements](#)^[1675]

3.6.2.4 Metrics

In Enterprise Architect, the Project Manager can allocate the possible weighting of defined metrics to work on a given model element.

To select the element to which to allocate metric weightings, click on the required element in the Project Browser.

Access [View | More Element Tools | Project Management > Metrics: New \(Ctrl+Shift+7\)](#)

Use to

- Enter the details of metrics that you might apply to work on an element

How to

To enter metric details for an element

Step	Action	See also
1	Select the element in the Project Browser.	
2	Select the Project Management menu option. The Project Management window displays, showing the Resource Allocation tab.	
3	Click on the Metrics tab.	
4	Click on the New icon on the Project Management window toolbar.	
5	On the Metrics tab enter the: <ul style="list-style-type: none"> • Name of the metric (a short description) • Type of the metric (Mandatory: type the name or click on the drop-down arrow and select; the selection list is drawn from the global Metric Type list, but any new metrics you type in the Type field are not added to the list) • Weighting allocated to the metric • Notes on the metric (this text is also displayed in the Notes window; it cannot be edited in that window) 	Metric Types ^[457] Notes ^[977]
6	Click on the Save icon in the Project Management window toolbar.	

Notes

- If you select a global metric type from the **Type** drop-down list and the associated **Weight** field is empty, the default Weight value is allocated to the **Weight** field on the Metrics tab
- To edit existing Metric items for this element, click on the required item in the:
 - List panel to the left of the window, in *Item* mode
 - List, in List mode
 - Project Management folder in the Element Browser window - if this window is not displayed, click on the **Browse Element** icon in the Project Management window toolbar; metric item icons have an **M** in the bottom right corner
- Although Enterprise Architect does not currently provide detailed reports on metrics within a model, you can use the Automation Interface or similar tools to create your own custom reports based on metric information you enter

Learn more

- [The Element Browser](#)^[876]
- [Enterprise Architect Object Model](#)^[2381]

3.6.2.5 Effort Types

Enterprise Architect enables you to add an effort *type* to the global list of effort types that can be added to any element in the model.

The global list of effort types displays in the **Type** field drop-down list on the Effort tab of the Project Management window.

Access **Settings | Project Types | Project Indicators > Effort: New**

Use to

- Specify the effort types used when assigning effort to an element

How to

To add a new effort type to the global list

Step	Action	See also
1	Select the Project Indicators menu option. The Project Indicators dialog displays.	
2	Click on the Effort tab.	
3	Click on the New button.	

Step	Action	See also
	(To edit an existing effort type, click on the effort type name in the Defined Effort Types list.)	
4	Complete the fields as follows: <ul style="list-style-type: none"> • In the Effort field type the name of the effort type • In the Description field type a short description of the effort type • In the Weight field type the default weighting to apply to the effort type • In the Note field, type any additional information on the effort type 	
5	Click on the Save button.	

Notes

- Although Enterprise Architect does not currently provide detailed reports on effort within a model, you can use the Automation Interface or similar tools to create your own custom reports based on effort information you enter
- You can transport effort types between models, using Export Reference Data and Import Reference Data

Learn more

- [The Project Management Window](#)^[448]
- [Enterprise Architect Object Model](#)^[238]
- [Export Reference Data](#)^[308]
- [Import Reference Data](#)^[308]

3.6.2.6 Metric Types

Enterprise Architect enables you to add a metric *type* to the global list of metric types that can be added to any element in the model.

The global list of metric types displays in the **Type** field drop-down list on the Metrics tab of the Project Management window.

Access [Settings | Project Types | Project Indicators > Metric: New](#)

Use to

- Specify the metric types used when assigning metrics to an element

How to

To add a new metric type to the global list

Step	Action	See also
1	Select the Project Indicators menu option. The Project Indicators dialog displays.	
2	Click on the Metric tab.	
3	Click on the New button. (To edit an existing metric type, click on the metric type name in the Defined Metrics list.)	
4	Complete the fields as follows: <ul style="list-style-type: none"> In the Metric Type field type the name of the metric type In the Description field type a short description of the metric type In the Weight field type the default weighting to apply to the metric type In the Note field, type any additional information on the metric type 	
5	Click on the Save button.	

Notes

- Although Enterprise Architect does not currently provide detailed reports on metrics within a model, you can use the Automation Interface or similar tools to create your own custom reports based on metric information you enter
- You can transport metric types between models, using Export Reference Data and Import Reference Data

Learn more

- [The Project Management Window](#)^[448]
- [Enterprise Architect Object Model](#)^[2387]
- [Export Reference Data](#)^[306]
- [Import Reference Data](#)^[308]

3.6.2.7 Risk Types

Enterprise Architect enables you to add a risk *type* to the global list of risk types that can be added to any element in the model.

The global list of risk types displays in the **Type** field drop-down list on the Risks tab of the Project Management window.

Access [Settings](#) | [Project Types](#) | [Project Indicators](#) > **Risk: New**

Use to

- Specify the risk types used when defining risks to an element

How to

To add a new risk type to the global list

Step	Action	See also
1	Select the Project Indicators menu option. The Project Indicators dialog displays.	
2	Click on the Risk tab.	
3	Click on the New button. (To edit an existing risk type, click on the risk type name in the Defined Risks list.)	
4	Complete the fields as follows: <ul style="list-style-type: none"> • In the Risk Type field type the name of the risk type • In the Description field type a short description of the risk type • In the Weight field type the default weighting to apply to the risk type • In the Note field, type any additional information on the risk type 	
5	Click on the Save button.	

Notes

- Although Enterprise Architect does not currently provide detailed reports on risks within a model, you can use the Automation Interface or similar tools to create your own custom reports based on risk information you enter
- You can transport risk types between models, using Export Reference Data and Import Reference Data

Learn more

- [The Project Management Window](#)^[448]
- [Enterprise Architect Object Model](#)^[2381]

- [Export Reference Data](#)^[306]
- [Import Reference Data](#)^[308]

3.6.3 The Project Information Window

The Project Information window documents tasks and issues that relate directly to the current project, and enables you to view and develop the project glossary of terms.

Access [View](#) | [More Project Tools](#) | [Project Information](#) (Alt+2)

Use to

Usage	See also
Review major project tasks that require attention; you can filter tasks based on their current status.	Project Tasks ^[460]
Review events, occurrences and situations that impact on project development and delivery.	Project Issues ^[462]
Review the technical and business terms already defined for a model. You can add to the list, delete or change items, and filter the list to exclude by type.	Project Glossary ^[468]

You can add or work on an item in the Project Information window by *right-clicking* (context menu) or *double-clicking* on the blank or completed item line.

Notes

- Right-clicking in the Project Information window displays a context menu that has options for filtering tasks/issues by status, and glossary by term; you can also rearrange the sort-order by clicking in the title bar of the column that the items are to be indexed on

3.6.3.1 Project Tasks

The *Project Tasks List* is a convenient 'To Do' list of major project work items that are not recorded elsewhere, and can be used to track events such as requests or corrections. It is a tab of the Project Information window.

Notes

- Right-click on the list to view the context menu, and select to add, modify or delete tasks, or to set a status filter
- To set the sort order, click the title-bar of the column on which to index the tasks
- To print out the currently displayed items, select the **Print List** context menu option
- You can transport task definitions between models using Export Reference Data and Import Reference Data

Learn more

- [Add, Modify and Delete Tasks](#)^[461]
- [Export Reference Data](#)^[306]
- [Import Reference Data](#)^[308]

3.6.3.1.1 Add, Modify and Delete Tasks

This topic explains how to maintain project tasks.

Access [View | More Project Tools | Project Information > Project Tasks](#)
[View | Project Calendar > Project Tasks mode](#)
[View | Personal Information > Project Tasks](#)

Use to

- Add project tasks
- Edit project tasks
- Delete project tasks

How to

To maintain tasks

Step	Action	See also
1	<p>If creating a new task:</p> <ul style="list-style-type: none"> • Double-click in a blank area of the Project Tasks tab, or on a cell of the Project Calendar <p>If editing an existing item:</p> <ul style="list-style-type: none"> • Double click on the item on the Project Tasks tab or Project Calendar <p>The Task Detail dialog displays.</p>	<p>The Project Information Window^[460]</p> <p>Project Calendar^[502]</p>
2	<p>Enter or update the following details of the task, selecting a value from the field's drop-down list where appropriate:</p> <ul style="list-style-type: none"> • The task name • Auto counters - if you have configured these, click on the Auto button to insert predefined auto-counter text; if you already have some text in the Name field, it is over-written by the auto-counter text • The task type • The task owner 	<p>Use Auto Naming and Auto Counters^[799]</p>

Step	Action	See also
	<ul style="list-style-type: none"> • The expected start and end date for the task (select the check boxes to activate the dates) • The current status of the task • The person this task has been assigned to • The task priority: high, medium or low • The expected total time for the task and the actual time expended • The percentage complete • The phase associated with this task • A description of the task • Any progress history appropriate to the task 	
3	Click on the Apply button.	
4	To create another entry, click on the New button, or to close, click on the OK button.	
5	To delete a task: <ul style="list-style-type: none"> • Right-click on the task and select the Delete context menu option • Click on the Yes button on the confirmation prompt 	

3.6.3.2 Project Issues

Any identified issues can be recorded against the current project, on either the:

- Project Issues tab of the Project Information window, or
- Project Issues dialog

For each Issue, you record the description, date, owner and status.

Access **View | More Project Tools | Project Information > Project Issues tab (Alt+2)**
Project | Documentation | Issues

Usage	See also
Add, edit and delete Issues	Add, Delete and Modify Issues ^[463]
Generate and view a rich text format report of your issue list	Report From Project Issues Dialog ^[465]

[Report From Project Issues Tab](#)^[466]

[Report Output Sample](#)^[466]

Notes

- You can transport these issue definitions between models, using Export Reference Data and Import Reference Data
- To print out the currently displayed items, select the **Print List** context menu option
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Issues** permission to update and delete Issues records

Learn more

- [Export Reference Data](#)^[306]
- [Import Reference Data](#)^[308]
- [List of Available Permissions](#)^[264]

3.6.3.2.1 Add, Delete and Modify Issues

This topic explains how to maintain project issue records using either the:

- Issue Detail dialog from the Project Issues tab of the Project Information window, or
- Project Issues dialog

These two dialogs have very similar fields.

Access **View | More Project Tools | Project Information > Project Issues tab (Alt+2), or Project | Documentation | Issues**

Use to

- Add project issues
- Edit project issues
- Delete project issues

How to

To maintain issues

Step	Action	See also
1	If creating a new issue: <ul style="list-style-type: none"> • On the Project Issues tab, double-click in a blank area to display the 	The Project Information Window ^[460]

Step	Action	See also
	<p>Issue Detail dialog</p> <ul style="list-style-type: none"> On the Project Issues dialog, click on the New button <p>If editing an existing item:</p> <ul style="list-style-type: none"> On the Project Issues tab, double click on the item On the Project Issues dialog, click on the item in the Project Issues & Discussion list 	
2	<p>Enter or update the following issue details, selecting a value from the field's drop-down list where appropriate:</p> <ul style="list-style-type: none"> The issue name Auto counters - if you have configured these, click on the Auto button to insert predefined auto-counter text; if you already have some text in the Name field, it is over-written by the auto-counter text The issue priority The date the issue was raised The issue status The issue owner A description of the issue The name of the person who resolved the issue The date on which the issue was resolved (select the check box to activate the date) Any comments on the resolution 	<p>Use Auto Naming and Auto Counters <small>[799]</small></p>
3	<p>Click on the Apply or Save button.</p>	
4	<p>If the issue is closed (and all the Resolution fields are completed), click on the Close Issue button.</p>	
5	<p>To create another entry, click on the New button, or to close, click on the OK or Close (Project Issues dialog) button.</p>	
6	<p>To delete an issue:</p> <ul style="list-style-type: none"> On the Project Issues tab, right-click on the entry and select the Delete context menu option, then click on the Yes button on the confirmation prompt On the Project Issues dialog, click on the item in the Project Issues & Discussion list, ensure that the Resolution fields are complete, then click on the Delete button 	

Step	Action	See also

Notes

- You cannot delete a Closed issue through the Project Issues dialog
- You can filter the list of issues by status, to include or exclude:
 - On the Project Issues dialog, closed issues, using the **Show Closed Issues** checkbox
 - On the Project Issues tab, all issues or just Open, Closed or Under Review issues, using the **Set Term Filter** context menu option

3.6.3.2 Report From Project Issues Dialog

This topic explains how to generate an RTF report on your project issue records using the Project Issues dialog.

Access [Project | Documentation | Issues](#)

Use to

- Generate and view an RTF report on project issues

How to

To generate your project issues report

Step	Action	See also
1	On the Project Issues dialog, click on the Report button. The Save As dialog displays.	
2	Browse for and select the appropriate file location. In the File name field, type the file name for the report.	
3	Click on the Save button. A status message displays when the report has been generated.	
4	Click on the OK button and then on the View RTF button. The report displays in your default viewer.	Report Output Sample [466]

3.6.3.2.3 Report From Project Issues Tab

This topic explains how to generate an RTF report on your project issue records using the Project Issues tab of the Project Information window.

Access [View | More Project Tools | Project Information > Project Issues tab \(Alt+2 \)](#)

Use to

- Generate and view an RTF report on project issues

How to

To generate your project issues report

Step	Action	See also
1	Right-click on the Project Issues tab; select the Create RTF Report context menu option. The Save As dialog displays.	
2	Browse for and select the appropriate file location. In the File name field, type the file name for the report.	
3	Click on the Save button. A status message displays when the report has been generated.	
4	Click on the OK button and then on the View RTF button. The report displays in your default viewer.	Report Output Sample [466]

3.6.3.2.4 Report Output Sample

The following illustration is an example of the output from an *Issues* report.

List of Project Issues: 24-Jul-2010 9:47:00 AM

Issue	Date/Owner	Description	Resolution
Test servers will be delayed	24/07/2010 Eloise Norman	The test server builds have been delayed because the particular (unusual) memory requirements to match the customer's site are not	Closed: 24/07/2010 Geoffrey Sparks The machines will be built and delivered using

		available on shore. They are being sourced from Singapore but it will delay the builds and delivery of the machines.	standard memory and the proprietary memory will be added later. All performance tests will be delayed until the memory is available.
Public Holidays	24/07/2010 Joanna Stoa	The schedule includes staff working on public holidays. A number of staff have indicated that contrary to what they stated earlier they are not available.	Open: 24/07/2010
Compiler Version disparity	24/07/2010 Eloise Norman	A number of the developers have downloaded different versions of a number of the compilers. This has lead to unpredictable builds impacting on testing.	Under Review: 24/07/2010

3.6.3.3 Project Glossary

The Project Glossary enables you to set up a list of defined terms for your project.



Access [View](#) | [More Project Tools](#) | [Project Information](#) > [Project Glossary](#) (Alt+2)
[Project](#) | [Documentation](#) | [Glossary](#)

Usage	See also
Add, delete and modify the project glossary entries through either the: <ul style="list-style-type: none"> Project Glossary tab on the Project Information window, or Glossary dialog 	Project Glossary Tab ^[468] The Glossary Dialog ^[469]
Separate glossary items by category; for example, Business terms and Technical terms.	Project Glossary Tab ^[468] The Glossary Dialog ^[469]
Save the glossary in Rich Text format for inclusion as part of a larger project document.	Generate a Report ^[471]
Create glossary terms and definitions from text in the Notes window, or from any Notes or Description fields that have the Notes toolbar.	Notes ^[971]
Insert existing glossary terms into any other Notes fields.	Notes ^[971]

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Glossary** permission in order to create, update or delete Glossary items; if security is not enabled, you can manage Glossary items without any permissions
- You might include a Glossary Report in your project requirements or functional specifications documents
- You can transport glossary definitions between models, using Export Reference Data and Import Reference Data

Learn more

- [Export Reference Data](#) 
- [Import Reference Data](#) 

3.6.3.3.1 Project Glossary Tab

The Project Glossary tab in the Project Information window shows displays the glossary terms already defined for your model.

Access **View | More Project Tools | Project Information > Project Glossary tab (Alt+2)**

Use to

- Add glossary terms
- Modify glossary terms
- Change a definition type for all terms of that type
- Filter the display show terms of a specific type only
- Print the displayed list of terms
- Delete glossary terms

How to

Right-click on an entry on the Glossary tab, and select the appropriate context menu option for the operation you intend to perform (as listed above); if you select the:

- **Add New** or **Modify Selected** options, the Glossary Detail dialog displays; complete the fields as indicated in the following table:

Field	Usage	See also
Term	Type the term to include in the glossary.	
Type	(Mandatory) Select the required type. If you require a different glossary type, click on the (...) (browse) button and specify the name of the new type. This field applies the type <i>only</i> to the selected term; you can rename a type for	

Field	Usage	See also
	all terms of that type, using the context menu as described below.	
Meaning	(Mandatory) Type the definition or description of the term. If necessary, format the text of this description using the Notes toolbar at the top of the field.	Notes
New	Clear the dialog fields so that you can define a new Glossary term.	
Apply	Save the new or updated glossary entry.	
Delete	Delete the entry. A prompt displays to confirm the deletion; click on the Yes button to remove the term from the glossary.	

- **Rename type** option, the Rename Glossary Type dialog displays on which you enter the alternative type name (either one of the existing types or a new type); when you click on **OK**, all entries of the original type are changed to the new type
- **Set term filter** option, the Term Type Filter dialog displays on which you enter the type of glossary term to list; when you click on **OK**, only items of that type are shown on the Project Glossary tab
- **Remove term filter** option, the Project Glossary tab refreshes to show glossary items of all types
- **Print List** option, the Print dialog displays on which you define the printing parameters; when you click on **OK**, the currently-displayed list prints out
- Delete option, a prompt displays to confirm the deletion; click on the **Yes** button to remove the term from the glossary

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Glossary** permission in order to create, update or delete Glossary items; if security is not enabled, you can manage Glossary items without any permissions

3.6.3.3.2 The Glossary Dialog

The Glossary dialog enables you to create and maintain definitions of terms used in your project, as a Glossary.

Access [Project | Documentation | Glossary](#)

Use to

- Add, modify and delete glossary entries
- Filter the dialog display to list entries of a specific type only

How to

Select the **Glossary** menu option and complete the Glossary dialog as indicated below:

Field	Usage	See also
Glossary Term	Type the term to include in the glossary.	
Glossary Type	(Mandatory) Select the required type. If you require a different glossary type, click on the (...) (browse) button and specify the name of the new type. This field applies the type <i>only</i> to the selected term; you can <i>rename</i> a type for all terms of that type ^[468] , using the Project Glossary tab of the Project Information window.	Project Glossary Tab ^[468]
Description	(Mandatory) Type the definition or description of the term. If necessary, format the text of this description using the Notes toolbar at the top of the field.	Notes ^[979]
Limit Display To	Select the appropriate glossary type to filter the Type Term list to show entries of that type only. Select All to display all glossary entries.	
New	Clear the dialog fields so that you can define a new Glossary term.	
Type Term	Review the list of defined glossary terms; click on an entry to edit it in the dialog fields.	
Save	Save the new or updated glossary entry.	
Delete	Delete the entry selected from the Type Term list.	
Report	Generate and print a glossary report.	Generate a Report ^[471] Glossary Report Output Sample ^[472]

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Glossary** permission in order to create, update or delete Glossary items; if security is not enabled, you can manage Glossary items without any permissions

3.6.3.3.3 Generate a Report

You can generate a rich text format file of your model's glossary, which you can either print or attach to any other appropriate document file.

The file can contain all glossary entries, or just those of selected types.

Access [Project | Documentation | Glossary: Report](#)

Use to

- Generate a formatted RTF document file of your model glossary

How to

To generate an RTF report of the project glossary

Step	Action	See also
1	Select the Glossary menu option. The Glossary dialog displays.	
2	Click on the Report button. The Glossary Report dialog displays.	
3	In the Filename field, type or select a filename for the glossary.	
4	In the Heading field, type a suitable heading for the glossary.	
5	In the Include Glossary Items panel, select the checkbox for each type of glossary entry to include. To select all types of entry, click on the Select All button.	
6	If necessary, to define the page setup and language for the report, click on the Page Setup and/or Language buttons.	
7	To include page breaks, select the Page break between sections checkbox.	

Step	Action	See also
8	Click on the Generate button to generate the report.	
9	Click on the View button to open the report.	Glossary Report Output Sample ^[472]

3.6.3.3.1 Glossary Report Output Sample

An example of the output from a Glossary report is shown below:

Glossary

Business Terms

Accounting Periods

A defined period of time whereby performance reports can be extracted. (normally 4 week periods).

Customer

A person or a company that requests An entity to transport goods on their behalf.

Technical Terms

Association

A relationship between two or more entities. Implies a connection of some type - for example one entity uses the services of another, or one entity is connected to another over a network link.

Component Model

The component model provides a detailed view of the various hardware and software components that make up the proposed system. It shows both where these components reside and how they inter-relate with other components. Component requirements detail what responsibilities a component has to supply functionality or behavior within the system.

Deployment Model

A model of the system as it is physically deployed.

Extends Relationship

A relationship between two Use Cases in which one Use Case 'extends' the behavior of another. Typically this represents optional behavior in a Use Case scenario - for example a user might optionally request a list or report at some point in a performing a business Use Case.

3.6.4 Project Task Allocation

The Project Task Allocation window enables you to review the allocation of work to elements in the project, focussing on either the elements that require work, or the resources required to perform the work. The window primarily shows information that is entered through other windows and dialogs, but once a record exists in the window you can edit it and, for example, add to or change the resources on an element.

Access [View | Task Allocations](#)

Topics

Topic	Detail	See also
Resource View	The Resource View shows the current commitments of each allocated resource in the project as a list of allocation records and a Gantt chart of the progress of the allocations.	Resource View 473
Element View	The Element View lists the elements in the project that have resources assigned to them, and the resources assigned to each element, as a list of allocation records and a Gantt chart of the progress of the allocations.	Element View 475
Report View	A Resource report shows how your resources are deployed in your project, displaying a list of all elements that have resources allocated to them.	Report View 477

3.6.4.1 Resource View

The Resource View of the Project Task Allocation window shows the current commitments of each allocated resource in the project as a list of allocation records and a Gantt chart of the progress of the allocations.

The display initially shows the resources and their overall commitment; click on the 'plus-box' to the left of the name of the resource to expand the entry to show the elements and the allocation period for each element.

The display shows both complete allocations and those that are still in progress; an internal filter hides completed allocations two weeks after the end date, and incomplete allocations one month after the end date.

Access [View | Task Allocations > Resource View](#)

Use to

- Check the dates on which specific resources or all resources are currently allocated
- Check the elements to which each resource is allocated
- Check the dates on which a resource is allocated to work on a specific element
- Check the general progress of the resource in completing the work
- Check the specific details of the allocation of a resource to an element and task or role

How to

On the Resource View tab, make selections from the options as indicated below:

Option	Usage	See also
Filter the display	<p>Right-click on the window and select the Apply Filter context menu option.</p> <p>The Filter Bar displays at the top of the Resource and Role or Task columns; type in a text string to filter for <i>element names</i> and <i>role or task</i> names, respectively, that begin with the text.</p>	List Header ^[590]
Display the properties of the element to which the resource is assigned	<p>Right-click on the entry and select the Object Properties context menu option.</p> <p>The Properties dialog for the element displays.</p>	Properties Dialog ^[843]
Display the resource allocation details	<p>Right-click on the entry and select the Task Properties context menu option.</p> <p>The Assigned Resources dialog displays; you can edit the details and, if necessary, change the resource allocated to the element.</p>	Resource Allocation ^[450] The Project Management Window ^[448]
Refresh the report	<p>Right-click on the entry and select the Refresh context menu option.</p> <p>The content of the display is refreshed and collapsed to resource level.</p>	
Display the Resource Allocation records for the element	<p>Right-click on the entry and select the Locate in Resource Window context menu option.</p> <p>The Project Management window opens at the Resource Allocation tab, with the details of the selected entry shown in the fields and the other resource allocations for the element listed in the left-hand panel.</p> <p>You can edit the details and, if necessary, change the resource allocated to the element.</p>	Resource Allocation ^[450] The Project Management Window ^[448]
Reposition the Gantt chart to show the start date or end date of an allocation, or today's date	<p>Right-click on the entry and select the required option:</p> <ul style="list-style-type: none"> • Go To Task Start • Go To Task End • Go To Today's Date <p>The display shifts to put the required date in the center of the chart.</p>	
Locate the element in the Project Browser	<p>Right-click on the element name and select the Find in Project Browser context menu option.</p> <p>The area of the Project Browser containing the element is brought into focus and expanded, and the element is highlighted.</p>	

Option	Usage	See also
Save an image of the Gantt chart to file	Right-click on the tab and select the Save Image to File context menu. option The Save As Image dialog displays, on which you specify the file name, location and graphics file type to save to.	
Save an image of the Gantt chart to the clipboard	Right-click on the tab and select the Copy Image to Clipboard context menu option. You can paste the image from the clipboard into your preferred graphics package.	

3.6.4.2 Element View

The Element View of the Project Task Allocation window lists the elements in the project that have resources assigned to them and the resources assigned to each element, as a list of allocation records and a Gantt chart of the progress of the allocations. An allocation can be partial, in that a role or task is assigned but no specific resource has been identified.

The display initially shows the elements and their overall resource commitment; click on the 'plus-box' to the left of the name of the element to expand the entry to show the resources and the allocation period for each resource.

The display shows both complete allocations and those that are still in progress; an internal filter hides completed allocations two weeks after the end date, and incomplete allocations one month after the end date.

Access [View | Task Allocations > Element View](#)

Use to

- Check the dates on which specific elements have resources currently allocated
- Check which resources are currently allocated to an element
- Assign further resources to the element
- Check the dates on which a resource is allocated to work on a specific element
- Check the general progress of the resource in completing the work
- Check the specific details of the allocation of a resource to an element and task or role

How to

On the Element View tab, make selections from the options as indicated below:

Option	Usage	See also
Filter the display	Right-click on the window and select the Apply Filter context menu option.	List Header ⁵⁹⁰

Option	Usage	See also
	<p>The Filter Bar displays at the top of the Element and Role or Task columns; type in a text string to filter for <i>resource names</i> and <i>role or task</i> names, respectively, that begin with the text.</p>	
<p>Display the properties of the element</p>	<p>Right-click on the element and select the Properties context menu option.</p> <p>The Properties dialog for the element displays.</p>	<p>Properties Dialog ^[843]</p>
<p>Display the resource allocation details</p>	<p>Right-click on the resource and select the Properties context menu option.</p> <p>The Assigned Resources dialog displays; you can edit the details and, if necessary, change the resource allocated to the element.</p>	<p>Resource Allocation ^[450]</p> <p>The Project Management Window ^[448]</p>
<p>Assign a new resource to the element</p>	<p>Right-click on the resource and select the Assign Resource context menu option.</p> <p>The Assigned Resources dialog displays; complete this as for the Resource Allocation tab of the Project Management window.</p>	<p>Resource Allocation ^[450]</p> <p>The Project Management Window ^[448]</p>
<p>Refresh the report</p>	<p>Right-click on the display and select the Refresh context menu option.</p> <p>The content of the display is refreshed and collapsed to element level.</p>	
<p>Display the Resource Allocation records for the element</p>	<p>Right-click on the entry and select the Locate in Resource Window context menu option.</p> <p>The Project Management window opens at the Resource Allocation tab, with the details of the selected entry shown in the fields and the other resource allocations for the element listed in the left-hand panel.</p> <p>You can edit the details and, if necessary, change the resource allocated to the element.</p>	<p>Resource Allocation ^[450]</p> <p>The Project Management Window ^[448]</p>
<p>Reposition the Gantt chart to show the start date or end date of an allocation, or today's date</p>	<p>Right-click on the entry and select the required option:</p> <ul style="list-style-type: none"> • Go To Task Start • Go To Task End • Go To Today's Date <p>The display shifts to put the required date in the center of the chart.</p>	

Option	Usage	See also
Locate the element in the Project Browser	<p>Right-click on the element name and select the Find in Project Browser context menu option.</p> <p>The area of the Project Browser containing the element is brought into focus and expanded, and the element is highlighted.</p>	
Save an image of the Gantt chart to file	<p>Right-click on the tab and select the Save Image to File context menu option.</p> <p>The Save As Image dialog displays, on which you specify the file name, location and graphics file type to save to.</p>	
Save an image of the Gantt chart to the clipboard	<p>Right-click on the tab and select the Copy Image to Clipboard context menu option.</p> <p>You can paste the image from the clipboard into your preferred graphics package.</p>	

3.6.4.3 Report View

The *Report View* shows how your resources are deployed in your project, displaying:

- A list of all elements that have resources allocated to them, and the type of each element
- The resource allocated, and the role played by that resource
- The start and end dates of the allocation
- The time allocated, expected and expended
- The percentage completion of the allocation

You can tailor the displayed information by:

- Hiding columns of information
- Grouping types of information
- Filtering the data by status
- Filtering the data by start date or end date
- Filtering any column to show only a specific value

Having displayed the information you require, you can print it.

Access [View | Task Allocations > Report View](#)


Use to

Identify, for example:

- What elements each resource is working on
- What resources are assigned to an element or package
- How close to completion your resources are, or
- How long a resource is assigned to a particular area.

How to

On the Report View tab, make selections from the options as indicated below:

Option	Usage	See also
Run the report	<p>Click on the first icon in the Report View toolbar (the rotating arrows).</p> <p>The report results display.</p> <p>If you have the report open for a while, you can update the information; either:</p> <ul style="list-style-type: none"> • Run the report again or • Right-click on the content and select the Refresh context menu option 	
Adjust column headings	<p>Drag and drop column headings into the sequence you require.</p> <p>Right-click on the column headings and select the Field Chooser context menu option, to add or remove columns using the Field Chooser dialog.</p>	
Group entries by column heading	<p>Right-click on the column headings and select the Enable Group Box context menu option, to cluster the report information according to your preferred hierarchy of column headings.</p>	List Header 
Filter columns	<p>Either:</p> <ul style="list-style-type: none"> • Click on the third toolbar icon from the <i>right</i> (the spy glass), or • Right-click on the column headings and select the Toggle Filter Bar context menu option <p>The filter field displays at the top of every column.</p> <p>Type in whatever text string you require in the appropriate column to filter the report to show only entries containing that text string in that column.</p>	
Filter by degree of completion	<p>In the first field in the toolbar, click on the drop-down arrow and select one of the following values:</p>	

Option	Usage	See also
	<ul style="list-style-type: none"> • All - Display all entries regardless of degree of completion • Completed - Display only those entries where the allocation is 100% completed • Above cut-off - Display only those entries that are more than a certain percentage complete • Below cut-off - Display only those entries that are less than a certain percentage complete <p>In the second field, either type a threshold value or increment to the value using the up/down arrows, to set the percentage completion for the Above/Below cut-off options.</p>	
Filter according to start/end date	<p>Click on the fourth toolbar icon from the right (the funnel). The Resource Filters dialog displays.</p> <p>In the Start Date and/or End Date field, click on the drop-down arrow and select the appropriate qualifier:</p> <ul style="list-style-type: none"> • After • Before • Equal To • Not Equals <p>In the date fields, click on the checkbox to activate the fields and either type in the day, month and year or click on the drop-down arrow to select the date from a calendar.</p> <p>The fields have an AND relationship; an entry must satisfy both date criteria before it is displayed.</p>	
Display the properties of the element to which the resource is assigned	<p>Right-click on the entry and select the Object Properties context menu option.</p> <p>The Properties dialog for the element displays.</p>	Properties Dialog [843]
Display the resource allocation details	<p>Right-click on the entry and select the Task Properties context menu option.</p> <p>The Assigned Resources dialog displays.</p>	Resource Allocation [450] The Project Management Window [448]
Display the Resource Allocation records for the element	<p>Right-click on the entry and select the Locate in Resource Window context menu option.</p> <p>The Project Management window opens at the Resource Allocation tab, with the details of the selected entry shown in</p>	Resource Allocation [450] The Project Management Window [448]

Option	Usage	See also
	the fields and the other resource allocations for the element listed in the left-hand panel.	
Locate the element in the Project Browser	<p>Right-click on the entry and select the Find in Project Browser context menu option.</p> <p>The area of the Project Browser containing the element is brought into focus and expanded, and the element is highlighted.</p>	
Print the report	<p>Either:</p> <ul style="list-style-type: none"> • Click on the second toolbar icon from the right (the printer) or • Right-click on an entry in the report and select the Print context menu option <p>The Print dialog displays, on which you specify the printer to use and the characteristics of the print job.</p>	

3.6.5 Spell Checking



Enterprise Architect provides a powerful spell checking facility. This operates at both project level and package level.

Topics

Topic	Detail	See also
Automatic Spell Checking	<p>The spell checker can be set to run automatically, so that it highlights possible errors in text as it is created or pasted in:</p> <ul style="list-style-type: none"> • Select Tools Options > Objects • Deselect or select the Disable spelling checkbox 	
Spelling Configuration	<p>You can configure what types of text error the spell checker should detect, using the Spelling Options dialog.</p>	Select Spell Checker Options ⁴⁸¹

Topic	Detail	See also
Using Languages Other Than English	Enterprise Architect is supplied with dictionaries for US English, Canadian English and British English; you can download a set of dictionaries for other languages from the Sparx Systems website.	Use Languages Other Than English ^[483]
Using the Spell Checker	You can run the spell checker manually on the whole model, or on a selected package.	Using the Spell Checker ^[485]
Correcting Text	As the spell check progresses, Enterprise Architect highlights any errors or unknown words; you have several options for responding to these potential spelling errors.	Correcting Words ^[486]

Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Spell Check** permission to perform any spell check or change any spell check options

Learn more

- [List of Available Permissions](#) ^[264]

3.6.5.1 Select Spell Checker Options

You can configure the spell checker to ignore or detect specific types of text conformation. Ignoring the conformation means to skip over the text without identifying it as an error.

Access [Tools](#) | [Spelling Tools](#) | [Spelling Options](#)

How to

To configure the spell checker, select the options for the checks you require on the Spelling Options dialog, as described below:

Option	Action	See also
Ignore capitalized words	Ignore any words beginning with a capital letter, such as <i>David</i> . You might enable this option if the text being checked contains many proper names.	
Ignore all-caps words	Ignore any words containing all capital letters, such as <i>RADAR</i> . You might enable this option if the text being checked contains many acronyms.	

Option	Action	See also
Ignore Words with Numbers	Ignore any words containing embedded digits, such as <i>Win95</i> and <i>Q4</i> . You might enable this option if the text being checked contains many code words or other symbols containing digits.	
Ignore Words with Mixed Case	Ignore any words containing an unusual mixture of upper- and lower-case letters, such as <i>MicroHouse</i> and <i>CapsLock</i> . You might enable this option if the text being checked contains many variable names or other symbols that use case changes to distinguish words.	
Ignore Domain Names	Ignore any words that appear to be Internet domain names, such as <i>wintertree-software.com</i> .	
Report Doubled Words	Detect any word appearing twice in a row, such as <i>the the</i> .	
Case Sensitive	Distinguish between capitalized and non-capitalized words. For example, <i>canada</i> would be considered as different from <i>Canada</i> , and therefore reported as a misspelling. When the option is disabled , <i>canada</i> and <i>Canada</i> are considered to be identical. Note that the performance of the spell checker is reduced if this option is disabled.	
Phonetic Suggestions	Suggest alternative words based on phonetic (sounds-like) similarity to the misspelled word; this option tends to improve the correction of badly misspelled words. Enabling this option increases the time required to locate suggestions. Note that either this option or the Typographical Suggestions option must be enabled in order to list suggestions of alternative words.	
Typographical Suggestions	Suggest alternative words based on typographical (looks-like) similarity to the misspelled word. This option is appropriate for people who are generally good spellers. Note that either this option or the Phonetic Suggestions option must be enabled in order to list suggestions of alternative words.	
Suggest Split Words	Suggest two separate words as a replacement for a misspelling containing two joined words. For example, <i>is the</i> would be suggested as a replacement for <i>isthe</i> .	

Option	Action	See also
Auto Correct	Automatically change words marked with <i>Auto Change</i> actions to their specified replacements. When this option is disabled , a confirmation prompt displays before the words are changed.	
Main Dictionary Language	Set the language of the main dictionary used to check spelling; the drop-down list shows only languages for which dictionaries are installed on your system. To check spelling in a different language, select the language in the list.	Use Languages Other Than English ^[483]
Suggestions	Determine the speed and accuracy of the initial automatic search for suggested replacements for misspelled words. Click on the appropriate radio button for your choice.	
OK	Close the Options dialog, and save any changes made to the option settings.	
Cancel	Close the Options dialog, and discard any changes made to the option settings.	

Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Spell Check** permission to perform any spell check or change any spell check options

Learn more

- [List of Available Permissions](#) ^[264]

3.6.5.2 Use Languages Other Than English

Enterprise Architect is supplied with three dictionaries, for US English, Canadian English and British English. Additional dictionaries are available as a set, for download from the registered pages of the Sparx Systems website.

The dictionaries in this downloadable file enable spell checking in the following languages:

- Brazilian
- Danish
- Dutch
- Finnish

- French (Continental)
- German
- Italian
- Norwegian
- Portuguese
- Spanish and
- Swedish

Use to

- Download and install the language pack of dictionaries so that you can select another language in which to perform the spell check

How to

To download the additional language dictionary pack

Step	Action	See also
1	Access the registered user page on the Sparx Systems web site.	Registered User page
2	At the end of the page, download the <i>EADict.zip</i> file from the Enterprise Architect Dictionary section.	
3	Unzip the file into the Enterprise Architect install directory - <i>C:\Program Files\Sparx Systems\EA</i> . This makes the non-English spelling dictionaries available to the Enterprise Architect spell checker.	Select Spell Checker Options ⁴⁸

Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Spell Check** permission to perform any spell check or change any spell check options
- Enterprise Architect .EAP files default to use Jet 3.5 as the database engine; this does not support unicode character sets, which makes the use of certain languages - such as Polish - difficult, including spell checking words in those languages

If you use such languages, it is recommended that you either upsize to a DBMS repository or set JET 4.0 as the database engine

If your .EAP project is not in a Jet 4.0 database, you should also download a copy of the Jet 4.0 EABase

model from the Sparx Systems website, and do an EAP to EAP transfer of your model into the Jet 4.0 file

Learn more

- [Server Based Repositories](#) ^[193]
- [General](#) ^[544]
- [EABase Jet4 page of the website](#)
- [Perform a Project Data Transfer](#) ^[442]
- [List of Available Permissions](#) ^[264]

3.6.5.3 Using the Spell Checker

Enterprise Architect has an inbuilt spell checker, which you can configure to detect a range of types of possible spelling error in your project, and to use one of several language dictionaries other than English.

Access [Tools | Spelling Tools | Spell Check Project Tools | Spelling Tools | Spell Check Current Package](#)

How to

To run the spell checker

Step	Action	See also
1	Select the Spell Check Project or Spell Check Current Package menu option, as required. The Spell Check dialog displays.	
2	(Spell Check Project option.) Select the checkbox against each of the items to spell check within your model.	
3	Click on the Start button to begin the spell check. As the spell check proceeds, the text being checked displays in the text panel at the bottom of the screen; if an error is detected, the Check Spelling dialog displays, offering several options to correct the error.	Correcting Words ^[486]

Notes

- The **Spell Check Project** menu option enables you to check spelling for the entire project
- The **Spell Check Current Package** option only checks the package currently open, and does not enable you to select items to check
- Enterprise Architect currently supports checking an entire model or a single package; a future release should support more detailed spell checking at the element and diagram level
- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of

Enterprise Architect, if security is enabled you must have **Spell Check** permission to perform any spell check or change any spell check options

Learn more

- [List of Permissions](#)

3.6.5.3.1 Correcting Words

As a spell check progresses, Enterprise Architect highlights any errors or unknown words in the Check Spelling dialog. The inbuilt spell check stores user-defined words in the user dictionary (*userdict.txt*) stored in the Enterprise Architect installation directory. During the spell check process, if you add a word, it is written into this file for later reference.

Use to

- Correct the spelling of a word
- Ignore the error
- Add the word to the user dictionary
- Suggest alternatives or
- Otherwise assist in the spelling correction process

How to

As the Spell Checker identifies possibly mis-spelled words, make the appropriate response:

Action	See also
<ul style="list-style-type: none"> • Modify the spelling by hand and click on the Change or Change All button to change the word to that spelling • Click on a suggested alternative and click on the Change or Change All button to change the word to that spelling • Click on the Ignore or Ignore All button to exclude the word from the spell check • Click on the Add button to add the word to the user dictionary • Click on the Suggest button to list alternative spellings or words • Click on the Cancel button to abort the spell check entirely 	

3.6.6 Personal Information

The Personal Information window enables you to record and manage your personal work within the project. The window operates on your identity as a defined Author on the project.

Access **View | Personal Information**

Topics

Topic	Detail	See also
Model Mail	You have the facility to send simple emails to registered project team members, and to receive messages from your team members.	Internal Mail ⁴⁸⁷
Allocated Work	Presents a Gantt chart on which you record the work that you are currently engaged in.	Recording Work ⁴⁹³
Project Tasks	Enables you to monitor and maintain the work tasks that have been assigned to you, or that you have created yourself.	Monitor Your Tasks ⁴⁹⁶
Workflow	Enables you to list and run your workflow scripts.	Monitor Workflow ⁴⁹⁸
Working Sets	Enables you to capture your current work area (diagrams and facilities currently open) as a working set for subsequent re-use.	Working Sets ⁴⁹⁸

3.6.6.1 Internal Mail

The *Model Mail* tab shows all emails that you have received from project team members within the project, under your User Security ID. You can open and respond to these emails and create new ones.

The *Sent Items* tab shows all the emails you have sent to project team members under your User Security ID. You can forward these emails and make further responses (see the final table in this topic).

This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, with User Security enabled; the mail system uses the individual and group users defined in User Security.

Access [View | Personal Information > Model Mail](#)
[View | Personal Information > Sent Items](#)

Use to

- Read messages from other project team members
- Read and respond to messages sent to you as a member of a group that holds a group mailbox
- Organize and sort both sent and received messages
- Create new messages, and reply to and forward received messages
- Delete messages from your In tray and Sent Items mailboxes

How to

On the Model Mail tab, review your messages as indicated below:

Option	Usage	See also
Open a message	Double-click on the message. If the message contains a hyperlink, click on the link to open or display the target file or object.	
Flag	Check the flag color to establish the significance of the message (the meanings of the flag colors should be defined within the project team).	
Status	Check for the read or unread icon; unread message items are anyway displayed in bold.	
Sender	The name of the project team member who sent the message.	
Subject	The topic of the message.	
Date	Indicates the age of the message, whether it was sent: <ul style="list-style-type: none"> • Today • Yesterday • This week • This Month • Last Month • Older 	
Sent	The exact date and time the message was sent.	
Select Columns	Right-click on the column headings and select the Field Chooser menu option, which enables you to add or remove specific columns from the display. You can also click on the column headings and drag them across the header bar to reposition the columns in a different sequence. Select the To column to display the names of the people to whom the email was sent; this is very useful to identify whether the email was sent to you as an individual or as a member of a group - if you received the email as a member of a group, remember that your actions on this single instance of the email affect all other recipients.	List Header [590]
Reorganize messages in the list	Either: <ul style="list-style-type: none"> • Click on the column heading and click on the arrow head to list items in order or reverse order, or • Right-click on the column head and select the Enable Group Box 	

Option	Usage	See also
	option to organize the messages into groups	

On the Model Mail tab, make selections from the toolbar options as indicated below:

Option	Usage	See also
Compose Message	Open the Model Message dialog, through which you create and send a mail message.	Create a Message [492]
Delete Messages	Delete a selected message or messages; you are prompted to confirm the deletion.	
Reply	Open the Model Message dialog, through which you create a response to the sender of the message, which includes a copy of the message and any earlier messages in the thread.	Create a Message [492]
Reply to All	Open the Model Message dialog, through which you create a response to the sender of the message and the other recipients, which includes a copy of the message and any earlier messages in the thread.	
Forward	Open the Model Message dialog, through which you forward the message to other project team members with, if necessary, your own comments.	
Mark as Unread	Mark the selected messages as unread.	
Mark as Read	Mark the selected messages as read.	
Viewing Panel	Select to: <ul style="list-style-type: none"> • Display the message text viewing panel underneath or to the right of the message list, or • Hide it so that you have to open the message to read it 	
Help Contents	Display this Help topic.	

On the Model Mail tab, make selections from the context menu options as indicated below:

Option	Usage	Shortcut	See also
Compose New Message	Click to open the Model Message dialog, through which you create and send a mail message.		
Reply	Create a response to the sender of the message, which includes a copy of the message and any earlier messages in the thread.	Ctrl + R	Create a Message [492]
Reply to All	Create a response to the sender of the message and the other recipients, which includes a copy of the message and any earlier messages in the thread.	Ctrl + Shift + R	
Forward	Forward the message to other project team members with, if necessary, your own comments.	Ctrl + F	
Set Message Flag	Select the appropriate flag color to establish the significance of the message (the meanings of the flag colors should be defined within the project team).		
Mark as Unread	Mark the selected messages as unread.	Ctrl + U	
Mark as Read	Mark the selected messages as read.	Ctrl + Q	
Delete	Delete a selected message or messages; you are prompted to confirm the deletion.		

On the Sent Items tab, manage the messages you have sent as indicated below:

Option	Usage	See also
Open a message	Double-click on the message. From the open message you can reply to the sender, reply to the sender and all other recipients, or forward the message to other team members.	Create a Message [492]
Flag	Check the flag color to establish the significance of the message (the meanings of the flag colors should be defined within the project team).	
To	Check the name(s) of the project team member(s) to whom the message was sent.	

Option	Usage	See also
Subject	The topic of the message.	
Date	Indicates the age of the message, whether it was sent: <ul style="list-style-type: none"> • Today • Yesterday • This week • Last Week • This Month • Last Month • Older 	
Sent	The exact date and time the message was sent.	
Select Columns	Right-click on the column headings and select the Field Chooser menu option, which enables you to add or remove specific columns from the display. You can also click on the column headings and drag them across the header bar to reposition the columns in a different sequence. Add the Sender column to your window; this is useful to identify whether you have responded to an email as an individual or as a member of a user group.	List Header [590]
Reorganize messages in the list	Either: <ul style="list-style-type: none"> • Click on the column heading and click on the arrow head to list items in order or reverse order, or • Right-click on the column head and select the Enable Group Box option to organize the messages into groups 	
Delete messages	Right-click on the message and select the Delete context menu option. You are prompted to confirm the deletion.	
Reply to All	Right-click on the message and select the Reply to All context menu option. The Model Message dialog displays.	Create a Message [492]
Forward	Right-click on the message and select the Forward context menu option. The Model Message dialog displays.	

3.6.6.1.1 Create a Message

The *Model Message* dialog enables you to compose messages to project team members within the project, under your Author ID.

Access [View | Personal Information > Model Mail](#)

Use to

- Create messages to other project team members
- Reply to messages from other team members
- Forward messages to other team members
- Link model components to the message, or add hyperlinks to files or other objects of relevance

How to

To create a message to another project team member as a new message, a reply to a message, or a forwarded message

Step	Action	See also
1	If your message or response concerns a diagram or model object, you could click on that object in the Project Browser now so that it is immediately available to be linked to your message.	
2	Select the appropriate toolbar icon, context menu option or message button to create, reply to or forward a message. The Model Message dialog displays.	Internal Mail 487
3	If you are creating or forwarding a message, or you want to send a reply to a wider audience, for <i>each</i> person you intend to send the message to: <ul style="list-style-type: none"> • Click on the To: button A list of project team member and group IDs displays • Scroll to the required name and double-click on it The name is added to the To: field 	
4	In the Subject field, type or edit the subject of the message as necessary.	
5	In the Flag field click on the drop-down arrow and on the appropriate flag color or option for your message.	
6	In the text panel, write the text of your message. You can format the text using the facilities of the Notes toolbar at the top of the field.	Notes Toolbar 979

Step	Action	See also
7	<p>If you intend to link a diagram or model object to this message, place the cursor at the appropriate position in your message text and click on the Insert Quick Link button.</p> <p>Select the object you highlighted in Step 1; a hyperlink to that object is inserted in the message text.</p>	
8	<p>If you did not select a model object and now intend to link to one, or you want to link to another target such as a file, Model Search or Team Review message, click on either:</p> <ul style="list-style-type: none"> • The Hyperlink icon in the toolbar or • The Insert Quick Link button and select the Other option <p>The Hyperlink Details dialog displays; create the required link.</p>	Hyperlinks <small>[1668]</small>
9	Click on the Send button to send the message to the recipients.	

Notes

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, with User Security enabled

3.6.6.2 Recording Work

The Allocated Work tab lists the elements to which *your* model Author ID has been allocated as a resource, where your Author ID is the same as:

- Your Enterprise Architect security user ID, if security has been enabled, or
- Your workstation login ID if security has not been enabled

For each element, the tab:

- In the left-hand panel lists the roles or tasks assigned to you as a resource on that element, for which the **Complete %** field value is less than **100**
- In the right hand panel displays a Gantt chart showing your progress in performing each role or task

You can add further work items for an element through the Allocated Work tab; however, you cannot delete any records. A record is no longer listed when the **Complete %** field value is **100**.

You or your supervisors can also add records through the Resource Allocation tab of the Project Management window.

Access [View](#) | [Personal Information](#) > [Allocated Work](#)

Use to

- View the elements to which you have been allocated as a resource, and the tasks you have been set to complete
- View and edit the properties of the elements you have been allocated to as a resource
- View and edit the details of each assigned role or task including, if you follow your project team procedures, re-assigning your task to another resource
- Add further tasks to an element
- Filter the display to show only items having a specific text element in the title, or restrict selected items to show just the element, not the tasks
- Locate the elements in the Project Browser
- Locate the tasks in the Resource Allocation tab of the Project Management window
- Print a list of tasks
- Capture an image of the display to a file or to the clipboard

How to

On the Allocated Work tab, make selections from the options as indicated below:

Option	Usage	See also
Review element properties	<p>Double-click on the element name.</p> <p>(Alternatively, right-click on the element name and select the Properties context menu option.)</p> <p>The Properties dialog for the element displays; review the pages as required.</p>	Properties Dialog [843]
Review resource task details	<p>Double-click on the task item.</p> <p>(Alternatively, either:</p> <ul style="list-style-type: none"> • Double-click on the progress bar for the item on the Gantt chart, or • Right-click on the task or role name and select the Properties context menu option) <p>The Assigned Resources dialog displays, which has the same content, format and functions as the Resource Allocation tab of the Project Manager window, in Item mode.</p> <p>Should it be necessary to reassign the item to another resource, click on the drop-down arrow on the Resource field and select the appropriate Author ID; when you save the changes, the item no longer appears in the list of tasks assigned to you.</p>	Resource Allocation [450] The Project Management Window [448]

Option	Usage	See also
Create new task item	<p>Right-click on the element name and select the Assign Resource context menu option.</p> <p>The Assigned Resources dialog displays, with your Author ID in the Resource field.</p> <p>Complete the dialog as for the Resource Allocation tab of the Project Manager window.</p>	Resource Allocation [450]
Refresh display to incorporate changes	<p>Your work item can be edited in a number of places in Enterprise Architect, such as the Project Task Allocation window and the Resource Allocation tab; the element to which it is assigned can be edited in these and many other areas.</p> <p>To refresh the display with any changes made elsewhere, right-click on the display and select the Refresh context menu option.</p>	Resource Allocation [450] Project Task Allocation [472] Properties Dialog [843]
Display the Resource Allocation tab of the Project Management window	<p>Right click on the list of items and select the Show Resources Window context menu option.</p> <p>The Resource Allocation tab of the Project Management window displays, listing all items for all resources associated with the element, and displaying the details for the first item in the list.</p>	Resource Allocation [450]
Locate the item in the Resource Allocation tab of the Project Management window	<p>Right-click on the work item and select the Locate in Resource Window context menu option.</p> <p>The Resource Allocation tab of the Project Management window displays, showing the item details for the selected item only.</p>	Resource Allocation [450]
Locate the element in the Project Browser	<p>Right-click on either the element or the work item and select the Find in Project Browser context menu option.</p> <p>The appropriate package hierarchy expands in the Project Browser, and the selected element is highlighted.</p>	
Expose hidden sections of the work item progress	<p>Some items might cover a long period of time, and you might not be able to display the complete progress line for the item in the Gantt chart.</p> <p>To locate the start point, expected end point, or today's date on the progress line for an item, right-click on either the item or the progress line and select the appropriate context menu option:</p> <ul style="list-style-type: none"> • Go To Task Start • Go To Task End • Go To Today's Date 	

Option	Usage	See also
	The Gantt chart shifts left or right to position the required point in the center of the display.	
Filter the work items	<p>You can refine the list of work items to show only those containing text that matches the filter item.</p> <p>Right-click on the tab and select the Apply Filter context menu option.</p> <p>The filter bar displays at the top of the panel; type in the filter text.</p> <p>As you type, the items listed and the Gantt chart are filtered to show only items where the item names match the text string.</p> <p>The filter does not operate on the element names.</p> <p>If you do not want to use the filter, right-click on the tab and select the Hide Filter Bar context menu option.</p>	
Capture an image of the Allocated Work data as a graphics file	<p>Right-click on the tab and select the Save Image to File context menu option.</p> <p>The Save As Image dialog displays, on which you specify the file name, location and graphics file type to save to.</p>	
Copy an image of the Allocated Work data to the clipboard	<p>Right-click on the tab and select the Copy Image to Clipboard context menu option.</p> <p>You can paste the image from the clipboard into your preferred graphics package.</p>	

3.6.6.3 Monitor Your Tasks

The Project Tasks tab lists the tasks that you either:

- Own, or
- Are assigned to

These tasks can be created through the:

- Project Tasks tab itself
- Tasks tab of the Project Information window
- Project Tasks view of the Resource Calendar

Access [View | Personal Information > Project Tasks](#)

Use to

- Review the status and progress of tasks that you are responsible for

- Create new tasks
- Update existing tasks
- Delete tasks
- Modify and filter the display of task information

How to

On the Project Tasks tab, make selections from the options as indicated below:

Option	Usage	See also
Add or Modify a task	You add or modify a task through the Task Detail dialog, which displays when you double-click on an entry (edit) or blank line (create).	Add, Modify and Delete Tasks ⁴⁶¹
Delete a task	Right-click on the message and select the Delete context menu option. You are prompted to confirm the deletion.	
Select columns	Right-click on the column headings and select the Field Chooser menu option, which enables you to add or remove specific columns from the display. You can also click on the column headings and drag them across the header bar to reposition the columns in a different sequence.	List Header ⁵⁹⁰
Reorganize tasks in the list	Either: <ul style="list-style-type: none"> • Click on the column heading and click on the arrow head to list items in order or reverse order, or • Right-click on the column headings and select the Enable Group Box option to organize the messages into groups <p>You can also use the filter bar to filter the display on an appropriate column value, such as the value New in the Status column; to display or hide the filter bar, right-click on the column headings and select the Toggle Filter Bar context menu option.</p>	List Header ⁵⁹⁰
Set persistent Status filter	Right click on the tab and select the Set term filter context menu option. The Filter by Status dialog displays, which enables you to select to list tasks of any status or only of one specific status. The filter you set persists when you close the Personal Information window or exit from Enterprise Architect.	

Option	Usage	See also
Print the task list	Right click on the tab and select the Print List context menu option. The Print dialog displays, on which you specify the local printer and the print characteristics.	

Learn more

- [Project Task Display](#)⁵¹³
- [Project Tasks](#)⁴⁶⁰

3.6.6.4 Monitor Workflow

The Workflow tab lists all of the workflow scripts that you have created:

You can execute any of the scripts listed on the tab.

Access [View | Personal Information > Workflow](#)

Use to

- Select and execute your workflow scripts

How to

On the Workflow tab, double-click on the required script to execute it.

Notes

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

Learn more

- [Workflow Scripts](#)²⁹⁸

3.6.6.5 Working Sets

The Working Sets tab lists the working sets that you have created.

A working set, rather like a project shortcut, is a group of diagrams and *views* that you have either captured from a work session or specifically drawn together, in either case using the Create Working Set dialog. It is a personal list, not visible to or active for any other user.

A *view* is:

- A Relationship Matrix profile to open

- The Team Review
- A Model Search to perform

Access [View | Personal Information > Working Sets](#)

Use to

- Capture your currently-open diagrams and views to resume work in the same configuration in a subsequent work session
- Create a working set by selecting diagrams, searches and profiles to open
- Open an existing working set to resume work in the required configuration (manually and automatically)
- Change a working set to remove, replace or add to the diagrams and views defined in it
- Delete a working set

How to

On the Working Sets tab, select the required option as indicated below:

Option	Usage	See also
Open Working Set	<p>Click on a working set name and either:</p> <ul style="list-style-type: none"> • Select the third icon in the toolbar or • Right-click and select the Open Working Set context menu option <p>Enterprise Architect opens all the diagrams and views listed in that set.</p> <p>If you already have a working set open, the 'new' set of views is added to the original set, with the last-listed diagram or view on top and in focus.</p> <p>You can open only one Model Search, Matrix profile or Team Review at a time; therefore, the last-opened Search, profile or Team Review replaces the previous instances.</p>	
Create New Working Set	<p>Either:</p> <ul style="list-style-type: none"> • Click on the first icon in the toolbar or • Right-click on the body of the tab and select the Create Working Set context menu option <p>The Create Working Set dialog displays.</p> <p>In the Name field, type a name for the working set.</p> <p>If you have diagrams and views <i>open</i>, the dialog lists them in the order in which they were opened; you can change the order using the '</p>	

Option	Usage	See also
	<p>hand buttons at the bottom of the dialog.</p> <p>Either select the checkbox against each diagram or view to include in the working set, or click on the Include All button.</p> <p>To add more diagrams or views that are not currently open, click on the Add Other button and select an option from the list:</p> <ul style="list-style-type: none"> • Add Diagram • Add Matrix Profile • Add Search • Add Team Review <p>You are prompted to select the required diagram, profile, search or Team Review topic; the view is then added to the list with its checkbox selected.</p> <p>Click on the OK button to create the working set.</p>	
<p>Build a Working Set from open diagrams or files</p>	<p>A variation on creating a working set is to generate the set from diagrams or files that are currently open.</p> <p>Right-click on the body of the tab and select the Create Working Set From context menu option, and either the Open Diagrams or Open Files sub-option as appropriate.</p> <p>The Create Working Set dialog displays.</p> <p>In the Name field, type a name for the working set.</p> <p>The dialog lists the diagrams and views that are currently open, in the order in which they were opened; with the checkboxes against the open diagrams or the open files already selected.</p> <p>You can either:</p> <ul style="list-style-type: none"> • Immediately click on the OK button to create the working set from the current environment, or • Change the content and selections as described for <i>Create New Working Set</i> (above) 	
<p>Add an open view or diagram to a working set</p>	<p>If, in your work, you open a diagram or view that you decide to add to a working set:</p> <ol style="list-style-type: none"> 1. Open the Working Sets tab. 2. Right-click on the required set name and select the Add Active View context menu option. <p>The open view is added to the working set.</p>	
<p>Duplicate a Working Set</p>	<p>If you want to use a variation of a working set without losing the original, you can create a copy and amend that copy as required.</p> <p>Right-click on the source working set and select the Copy as New</p>	

Option	Usage	See also
	<p>context menu option.</p> <p>The Create Working Set dialog displays.</p> <p>In the Name field, type a name for the working set.</p> <p>Each of the listed diagrams and views is already selected; make any changes you require, and click on the OK button to save the list as a new working set.</p>	
Edit a working set	<p>To change the diagrams and views in a working set, or change the set name, click on the set name and either:</p> <ul style="list-style-type: none"> • Click on the second icon in the toolbar or • Right-click and select the Edit context menu option <p>The Create Working Set dialog displays.</p> <p>Make any changes you require, and click on the OK button to save the changes.</p>	
Delete Working Set, or View in Working Set	<p>Expand the working set and select a specific item (to remove that item) or the set name (to move the whole set)</p> <p>Either:</p> <ul style="list-style-type: none"> • Click on the fifth icon in the toolbar or • Right-click on the set and select the Delete Working Set Item context menu option <p>A prompt displays to confirm the deletion; click on the Yes button.</p> <p>The item or set is removed.</p> <p>If you delete a working set that you have opened, the diagrams and views from that set remain open until you specifically close each one or close Enterprise Architect.</p>	
Automatically apply the working set when the model is opened	<p>Right-click on the working set name and select the Apply when model opens context menu option.</p> <p>This option is a toggle that turns the same option off for the previous working set on which it was selected.</p> <p>The folder turns green, indicating that the option is on for this working set.</p> <p>When you close the model and subsequently open it, the diagrams and views in the working set are automatically opened.</p>	
Track working set/ views last opened	<p>You might use several working sets in a work session, and close some views and open others.</p> <p>To capture the diagrams and views you actually have open when you</p>	

Option	Usage	See also
	<p>close down Enterprise Architect, right click on the body of the tab and select the Store Main Tab View History context menu option. A new, empty working set folder displays at the top of the list - <i>Windows open when Model was last closed</i>.</p> <p>When you reopen the model in a new work session, this new folder is automatically populated with a list of the diagrams and views that were open when you last closed the model; this is a read-only set that cannot be edited.</p> <p>You can expand the set and double-click on specific windows to re-open them, or right-click on the set and use the Open Working Set option to reopen all the diagrams and views you had open before.</p> <p>However, it is recommended that you use the Apply when model opens context menu option on the set; thereafter, whenever you open the model, all the diagrams and views you had open when you closed the model are automatically re-opened.</p>	
<p>Locate item in Project Browser</p>	<p>To find a diagram from your working list in the Project Browser, right-click on the diagram name and select the Find in Project Browser context menu option.</p> <p>The section of the Project Browser containing the diagram is expanded, and the diagram name is highlighted.</p>	

Learn more

- [Create Project Shortcut](#)^[185]
- [Capture Current Work Environment](#)^[187]


3.6.7 Project Calendar

The Project Calendar provides a project management overview in calendar format of the deployment of resources, timeframes for tasks, and upcoming project events such as meetings and milestones. The calendar displays in the Diagram View workspace, consisting of a:

- Calendar panel, displaying months of the year
- Diary panel sectioned into days or weeks
- Toolbar at the top of the window, which enables you to define what information is displayed and how it is formatted

Access [View | Project Calendar](#)

Topic	Detail	See also
<p>Calendar Panel</p>	<p>The calendar panel, when viewed normally on the screen, displays a calendar of six months that by default includes the current month.</p> <p>By closing or reducing other panels on the screen, and depending on the size of your screen, you can show the</p>	

Topic	Detail	See also
	<p>months for a period of up to 3 years and six months.</p> <p>You can review data for up to several decades into the past, if such data has been recorded, and plan ahead for up to several decades into the future; to display future or past months, either:</p> <ul style="list-style-type: none"> • Click on the arrows in the names of the months on the left and right of the top row of months, or • Click and hold the mouse on the name of a month so that a list of months and years displays, then move the mouse forwards to scroll up the list or backwards to scroll down the list; when you reach the required month, ensure that it is highlighted and release the mouse button <p>On the calendar, today's date has a red border; if you have scrolled the calendar so that today's date is not shown, click on the second icon in the toolbar () to position the current month in the top left of the calendar with today's date highlighted.</p>	
Diary Panel	<p>The diary panel displays a set of day fields, each of which can contain one or more markers for an event or task.</p> <p>The content of the display is determined by the selections you make in the toolbar, and principally by the object type you select from the first drop-down field:</p> <ul style="list-style-type: none"> • Project Calendar • Allocated Resources • Project Tasks <p>The structure of the display depends on the period you want to review, specified using the icons on the toolbar:</p> <ul style="list-style-type: none"> • 1 day (Project Calendar only) • 5 days (Mon - Fri) (Project Calendar only) • 7 days (Mon - Sun) • 31 days <p>However, in the Calendar panel you can highlight specific periods of between 1 and 7 days (Project Calendar only) and 14, 21, 28, 35 and 42 days, to display just those periods in the Diary panel.</p> <p>If you select a period of between 1 and 7 days, each day is divided into one hour time intervals; you can adjust the time intervals for these periods, using the toolbar (see below).</p>	<p>Calendar^[505]</p> <p>Allocated Resources^[510]</p> <p>Project Tasks^[460]</p>

Topic	Detail	See also
	<p>For the current day, the current time is also highlighted.</p> <p>Also, if you click on today's date in the Calendar panel the Diary panel changes to show the shortest period (7 days or 1 day) for the object type, again with adjustable time intervals for the times of day.</p> <p>For displays of 7 or more days, you can scroll up or down to display the information for earlier or later dates; if you scroll away from today's date, you can return to it by right-clicking anywhere on the display and selecting the Show Today context menu option (not Project Calendar).</p>	
Toolbar	<p>The toolbar options modify what is displayed on the Diary panel, and enable you to add certain kinds of information.</p> <p>The icons and fields in the toolbar, from left to right, have the following functions:</p> <ul style="list-style-type: none"> • Add New <object> - enables you to create a new record for an event or task • Assign Resource - for Allocated Resources, enables you to assign a new resource to the task • Show Today - re-focuses the Diary panel and Calendar panel on today's date, for any object type • 1, 5, 7, 31 - sets the number of days displayed in the Diary panel • First drop-down - identifies the type of information to display: Calendar events, allocated Resources, or Tasks • Second drop-down - identifies the type or characteristic of object to filter for (and depends on the value selected in the first field) • Third drop-down - for Allocated Resources, identifies the element type to filter for • Options: <ul style="list-style-type: none"> • Configure Event Subtypes - Displays the Configure Subtype dialog, which you use to define categories of event to record on the Calendar • Time Scale - where the display includes times of day, enables you to reset the time interval • Show End Time - for the Project Calendar, where the times of day are NOT shown and an event occurs within one day, this shows or hides the 	<p>Configure Event Subtypes^[509]</p>

Topic	Detail	See also
	<p>time at which the event finishes (the start time displays automatically)</p> <ul style="list-style-type: none"> • Show Time As Clocks - toggle between showing start and end times in digital format and as a clock face • Compress Weekend Days - in 31-day format, toggle between showing Saturday and Sunday as separate fields and a pair of half-fields • Show ToolTips - toggle between showing and hiding mouse roll-over tooltips 	

Notes

- The Project Calendar is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- In the Lite edition of Enterprise Architect, you can view Calendar entries but not change them

3.6.7.1 Calendar

Project Calendar is a display mode of the project Calendar facility. In Project Calendar mode, the Diary panel shows the selected period of time (see the *Project Calendar* topic) containing flags for each *event* that takes place in or through that period. These events can be, for example, meetings, staff absences or commitments, product releases or public holidays.

You can customize the range of events that you record, using the Configure Subtype dialog.

Access **View | Project Calendar > Project Calendar**

How to

To record an event in the Project Calendar

Step	Action	See also
1	Double click on the required day, or the first day in a longer period. The Project Event dialog displays.	
2	In the Subject field, type the name of the event.	
3	If the event is to take place at a specific place, in the Location field type the name of the place or room.	

Step	Action	See also
4	<p>In the Category field, click on the drop-down arrow and select the appropriate categorization:</p> <ul style="list-style-type: none"> • Event (such as an external trade show, or internal presentation) • Meeting (whether internal or external) • Milestone/Objective (such as a product test or release) • People (an absence or commitment of a specific staff member) 	
5	<p>If the event is likely to occupy one complete day, select the All day event checkbox; this:</p> <ul style="list-style-type: none"> • Places the event icon at the top of the day field in the Diary, above any time sections • Hides the time sections of the Start time and End time fields 	
6	<p>In the Start time and End time fields, specify the start and end dates and/or times of the event.</p> <p>In the date section of each field, either:</p> <ul style="list-style-type: none"> • Click on the drop-down arrow to display the current month and select the month and date, or • Click on the 'spin' arrows to roll the date forwards or backwards by one day at a time <p>In the time section of each field, click on the hour or minute components (which are separate) and either:</p> <ul style="list-style-type: none"> • Type in the required time or • Click on the 'spin' arrows to roll the time backwards or forwards 	
7	<p>In the Event Type field, click on the drop-down arrow and select an appropriate type label for the event; the options change depending on the value you selected for the Category field.</p> <p>This provides the fill color for the event icon, as indicated by the fill box next to each option.</p>	
8	<p>In the Defined as field, click on the drop-down arrow and select the indicator for whether the event is internal or external to your organization.</p>	
9	<p>If the event consists of a telephone call, select the Phone Call checkbox.</p>	
10	<p>In the description field, type any notes required on the event; these display in the mouse roll-over tool tip for the event on the Diary panel.</p>	

Step	Action	See also
11	If this event repeats at regular intervals, click on the Recurrence button and complete the Event Recurrence dialog (see below).	
12	Click on the OK button to save the event and display it on the Calendar.	

Recurring Events:

To define a recurring event on the Event Recurrence dialog, follow the steps below (some information defaults from the Project Event dialog, but you might have to edit it to accurately define the recurrence details):

Step	Action	See also
1	In the Start and End fields, specify the start and end times of the event; click on the hour or minute components (which are separate) and either: <ul style="list-style-type: none"> Type in the required time or Click on the 'spin' arrows to roll the time backwards or forwards 	
2	In the Duration field, click on each of the day(s), hours and minutes portions and either type or 'spin' to the required value. The hours and minutes portions are linked to the Start and End fields, so that: <ul style="list-style-type: none"> As you edit the Start field or the Duration field, the End field adjusts to maintain the duration relative to the start time As you edit the End field, the Duration field changes to match the difference between the start and end times 	
3	In the Recurrence pattern panel, select the appropriate radio button for the frequency of the event. As you select the radio button, the fields on the right of the panel change to further define when the event recurs.	
4	Select the appropriate radio buttons and field values, with the following guidance: <ul style="list-style-type: none"> <i>Daily</i> can be every 1, 2, 3, 4, 5 or 6 days, or every day of the working week <i>Weekly</i> is on one or more specific days of the week, the events being separated by a period of between 1 and 51 complete weeks <i>Monthly</i> can be on a specific date or a specific day of the month, the events being separated by a period of between 1 and 11 complete months <i>Yearly</i> can be on a specific date or a specific day of a specific month 	

Step	Action	See also
5	<p>In the Range of recurrence panel, in the Start field, specify the date on which the event cycle begins.</p> <p>Click on the day, month and year components (which are separate) and either:</p> <ul style="list-style-type: none"> • Type in the required value or • Click on the 'spin' arrows to roll the date backwards or forwards 	
6	<p>Select the appropriate radio buttons to indicate that the recurrence cycle:</p> <ul style="list-style-type: none"> • Has no defined end point • Ends after a specific number of occurrences of the event (type in the number) • Ends by a specific date (enter the date) 	
7	<p>Click on the OK button, and again on the Project Event dialog.</p> <p>The event icon displays at all points in the Calendar where it is scheduled to occur, as defined in the Event Recurrence dialog; the recurrence is indicated by a graphic depicting two recirculating arrows.</p>	
8	<p>If the recurrence is no longer required:</p> <ul style="list-style-type: none"> • Double-click on the icon for any occurrence of the event, to display the Project Event dialog • Click on the Recurrence button to display the Event Recurrence dialog • Click on the Remove Recurrence button; the Event Recurrence dialog closes • Click on the OK button; the Project Event dialog closes <p>Only the first occurrence of the event remains in the Calendar.</p>	

Notes

- The Project Calendar is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- In the above editions, if security is enabled you must have **Manage Project Calendar** permission in order to create, update or delete Project Calendar events; if security is not enabled, you can change data without this permission
- In the Lite edition of Enterprise Architect, you can view Calendar entries but not change them

Learn more

- [Project Calendar](#)^[502]
- [Configure Event Subtypes](#)^[509]

- [List of Available Permissions](#) ²⁶⁴

3.6.7.1.1 Configure Event Subtypes

The Configure Subtype dialog enables you to define additional types of event to record on the Project Calendar.

Access [View | Project Calendar:](#)  [Configure Event Subtypes](#)

Use to

- Define a new event type within a select event category
- Change an existing event type
- Delete an existing event type

How to

To manage the event types for your project

Step	Action	See also
1	On the Configure Subtype dialog, in the Category field, click on the drop-down arrow and select the appropriate category for the event.	
2	Click on the Type field and type a suitable name for the event type (or select an existing event from the list below the Category field).	
3	In the Color field, click on the drop-down arrow and select or define a color for the event icon.	
4	Click on the Save button to save the new or edited event type to the list for the specified category. Click on the New button if you are going to create another event type.	
5	If the event type is no longer required, click on the Delete button.	
6	If required, you can change the sequence of the event types in the displayed list, using the 'Up Hand' and 'Down Hand' buttons at the bottom of the dialog.	
7	Click on the OK button to close the dialog; any new event types are available for use in the Project Calendar.	

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Project Calendar** permission in order to create or delete Project Calendar event subtypes; if security is not enabled, you can change data without this permission

Learn more

- [Calendar](#)^[505]

3.6.7.2 Allocated Resources

Allocated Resources is a display mode of the project Calendar facility.

In Allocated Resources mode, the Diary panel shows the selected period of time (see the *Project Calendar* topic), with at least one icon representing each element in the project to which a resource has been allocated during that period. If an element has more than one allocated resource, each element:resource combination is represented separately.

Each icon shows the element name and resource name, and indicates the status of the assigned work with one of the following symbols:

- a green square, indicating that the resource has been assigned the work
- a green tick, indicating that the resource has completed the assigned work
- a red square, indicating that the resource has not completed the assigned work
- a square divided diagonally with green and red halves, indicating that the assigned work is to be completed within one working day

Typically, while the work is in progress a resource is represented by:

- An icon with a green square on the day the work is planned to start, and
- An icon with a red square on the day the work is planned to finish

When the work is complete, the icons on both days display the green tick.

Resources are initially allocated to elements through the:

- Resource Allocation tab of the Project Management window, or
- Allocated Work tab of the Personal Information window

You can then *edit* these allocations through either of those windows or through the Allocated Resources Calendar itself.

Access [View | Project Calendar > Allocated Resources](#)

Use to

- Review the allocation of resources across the project over a period of time

- Review the status of the work assigned to those resources
- Display and edit the details of the task assigned to a resource
- Allocate further resources to an element
- Delete a resource allocation from an element
- Create and send an email to the resource allocated to an element

How to

On the Allocated Resources Calendar, make selections from the options as indicated below:

Option	Usage	See also
Display/edit details of an allocated resource	<p>Either:</p> <ul style="list-style-type: none"> • Double-click on any icon for that element:resource combination, or • Right-click on the icon and select the Properties context menu option <p>The Assigned Resources dialog displays, showing the details of the assignment of the resource to a task on the element.</p> <p>This dialog has the same content, format and functions as the Resource Allocation tab of the Project Manager window, in Item mode.</p>	Resource Allocation ^[450] The Project Management Window ^[448]
Add resource to element	<p>Right-click on any icon for the element and select the Assign Resource to <element name> context menu option.</p> <p>The Assigned Resources dialog displays, showing the start and end dates both set to the date from which you selected the icon, and the Allocated Time field set to 1; all other fields are blank.</p> <p>This dialog has the same content, format and functions as the Resource Allocation tab of the Project Manager window, in Item mode.</p> <p>Specify the resource and the role or task that resource is performing, and define the period for which the resource is allocated to the element.</p>	Resource Allocation ^[450] The Project Management Window ^[448]
Message Resource	<p>Right-click on any icon for the element:resource combination and select the Message Resource <resource name> context menu option.</p> <p>The Model Message dialog displays, on which you create and send your message to the resource allocated to the element.</p>	Create a Message ^[492]
Delete resource from element	<p>Right-click on any icon for the element:resource combination, and select the Delete Resource from <element name> context menu option.</p>	

Option	Usage	See also
	<p>A prompt displays to confirm the deletion; click on the Yes button.</p> <p>The icon and any corresponding icons for that element: resource combination are deleted from the calendar, and the resource is no longer allocated to that element.</p>	
Locate element in diagrams in which it is used	<p>Right-click on any icon for the element, and select the Find in all Diagrams context menu option.</p> <p>If the element is used in only one diagram, that diagram displays.</p> <p>If the element is used in more than one diagram the Element Usage dialog displays, listing the diagrams in which the element occurs.</p> <p>Select the required diagram and click on the Open button to display that diagram.</p> <p>This option also operates on Port and Part Property Type Classifiers.</p>	Show Element Use 802
Locate element in Project Browser	<p>Right-click on any icon for the element, and select the Find in Project Browser context menu option.</p> <p>The area of the Project Browser containing the element is brought into focus and expanded, and the element is highlighted.</p>	
Locate resource allocation start date corresponding to allocation end date	<p>Firstly, click on the end date icon for the element:resource combination to highlight it; the corresponding start date icon is also highlighted.</p> <p>If the start date icon is not in view, right-click on the end date icon and select the Show Start/End context menu option; the display scrolls to show the start date and the start date cell is highlighted.</p>	
Locate resource allocation end date corresponding to allocation start date	<p>Firstly, click on the start date icon for the element:resource combination to highlight it; the corresponding end date icon is also highlighted.</p> <p>If the end date icon is not in view, right-click on the start date icon and select the Show Start/End context menu option; the display scrolls to show the end date and the end date cell is highlighted.</p>	
Refocus display on today's date	<p>Right-click anywhere on the display and select the Show Today context menu option.</p> <p>The display returns to today's date, which is highlighted and outlined.</p>	

Notes

- The Project Calendar is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- In the above editions, if security is enabled you must have **Manage Project Information** permission in order to allocate resources to or remove resources from the project Calendar; if security is not enabled, you can change data without this permission
- In the Lite edition of Enterprise Architect, you can view Calendar entries but not change them
- As the start and end icons for a *completed* task are identical, and one-day tasks have only one icon, the methods to show the corresponding start date or end date for an icon are also useful to show whether the icon *is* for a start or end date, or for a one-day task; alternatively, double-click on the icon and review the allocation dates

Learn more

- [Project Calendar](#)^[502]
- [List of Available Permissions](#)^[264]

3.6.7.3 Project Tasks

Project Tasks is a display mode of the project Calendar facility.

In Project Tasks mode, the Diary panel shows the selected period of time (see the *Project Calendar* topic), with at least one icon representing each *project task* scheduled during that period. The icon represents an independent record of the task - there are no comparisons or validations of the tasks. Therefore you can have separate icons for the same task, with different resources allocated or no resources allocated; you can even have identical task records, if these serve a purpose for you.

Each icon shows the task name and resource name, and indicates the status of the task with one of the following symbols:

- a green square, indicating that the task has been scheduled
- a green tick, indicating that the task is complete
- a red square, indicating that the task is incomplete
- a square divided diagonally with green and red halves, indicating that the task is to be completed within one working day

Typically, while a task is in progress it is represented by:

- An icon with a green square on the day the work is planned to start, and
- An icon with a red square on the day the work is planned to finish

When the work is complete, the icons on both days display the green tick.

Tasks can be created and managed through this view of the Project Calendar, or the:

- Project Tasks tab of the Project Information window, or

- Project Tasks tab of the Personal Information window

Access [View | Project Calendar > Project Tasks](#)

Use to

- Review the tasks across the project over a period of time
- Review the progress of those tasks
- Create new tasks
- Display and edit the details of the task
- Allocate a resource to a task
- Delete a task
- Create and send an email to the resource assigned to a task
- Create and send an email to the owner of the task

How to

On the Project Tasks display, make selections from the options as indicated below:

Option	Usage	See also
Create a task	<p>Either:</p> <ul style="list-style-type: none"> • Double-click on the date cell in which the task is to start • Right-click on the cell and select the Add New Task context menu option, or • Click on the down-arrow next to the first icon in the Calendar toolbar, and select the New Task option <p>The Task Details dialog displays, through which you create the task.</p>	Add, Modify and Delete Tasks ^[46]
Display and edit a task	<p>Either:</p> <ul style="list-style-type: none"> • Double-click on the task • Right-click on the cell and select the Properties context menu option, or • Click on the down-arrow next to the first icon in the Calendar toolbar, and select the Properties option <p>The Task Details dialog displays; if necessary, edit the information.</p>	Add, Modify and Delete Tasks ^[46]
Allocate a	Certain tasks might not initially be assigned to a resource, but	

Option	Usage	See also
resource to a task	might later require specific assignment to finish them off. Open the task to edit it, click on the drop-down arrow in the Assigned field and select the required resource.	
Delete a task	Right-click on any icon for the required task and select the Delete '<task-name>' context menu option. You are prompted to confirm the deletion; click on the Yes button.	
Create and send a message to the task owner	Right-click on any icon for the task and select the Message Owner context menu option. The Model Message dialog displays, on which you create and send your message to the user who owns the task (as identified in the Task Details dialog). If the task does not have a defined owner, this option does not operate.	Create a Message [492]
Create and send a message to the resource assigned to the task	Right-click on any icon for the task and select the Message Assigned <resource name> context menu option. The Model Message dialog displays, on which you create and send your message to the resource assigned to complete the task (as identified in the Task Details dialog). If the task does not have a defined resource, this option does not operate.	
Locate task start date corresponding to task end date	Firstly, click on the end date icon for the task to highlight it; the corresponding start date icon is also highlighted. If the start date icon is not in view, right-click on the end date icon and select the Show Start/End context menu option; the display scrolls to show the start date and the start date cell is highlighted.	
Locate task end date corresponding to task start date	Firstly, click on the start date icon for the task to highlight it; the corresponding end date icon is also highlighted. If the end date icon is not in view, right-click on the start date icon and select the Show Start/End context menu option; the display scrolls to show the end date and the end date cell is highlighted.	
Refocus display on today's date	Right-click anywhere on the display and select the Show Today context menu option. The display returns to today's date, which is highlighted and outlined.	

Notes

- The Project Calendar is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- In the Lite edition of Enterprise Architect, you can view Calendar entries but not change them

Learn more

- [Project Calendar](#) ^[502]
- [Project Tasks](#) ^[460]
- [Monitor Your Tasks](#) ^[496]

3.6.8 Use Case Estimation

Project estimation is the task of working out how much time and effort is required to build and deploy a solution.

The Use Case metrics facility in Enterprise Architect provides a starting point for estimating project effort; using this facility you can get a rough measure of the complexity of a system and some indication of the effort required to implement the model. Like all estimation techniques, Use Case metrics requires some experience with previous projects to 'calibrate' the process.

There is additional information available on Use Case metrics on the Sparx Systems website.

Topics

Topic	Detail	See also
Calibrating	<p>The following values must be carefully calibrated in order to gain the best possible estimates:</p> <ul style="list-style-type: none"> • Technical Complexity Factors, which are values that attempt to quantify the difficulty and complexity of the work in hand • Environment Complexity Factors, which are values that attempt to quantify non-technical complexities such as team experience and knowledge • Default Hour Rate, which sets the number of hours per Use Case point 	<p>Technical Complexity Factors ^[517]</p> <p>Environment Complexity Factors ^[518]</p> <p>Default Hours ^[520]</p>
Estimating	<p>Once you have entered all the calibration values, you can estimate the project timescale through the Use Case Metrics dialog.</p>	<p>Estimating Project Size ^[520]</p>

Learn more

- [Project Estimation Using Use Case Metrics](#)

3.6.8.1 Technical Complexity Factors

Technical Complexity Factors (TCFs) are used in the Use Case Metrics estimation technique.

The EABase.eap model contains a default set of TCFs, which you can add to or modify using the Estimation Factors dialog. This set of factors should include all factors that could affect the technical complexity of the project environment.

Access [Settings](#) | [Project Types](#) | [Estimation Factors](#) > [Technical Complexity Factors](#)

How to

To maintain Technical Complexity Factors

Step	Action	See also
1	On the Technical Complexity Factors tab, either: <ul style="list-style-type: none"> Click on the New button to add another TCF, or Click on the required factor in the Defined Technical Types list to edit it (go to step 3) 	
2	In the Factor Number field, type the appropriate TCF number.	
3	In the Description field, type or edit the TCF description.	
4	In the Weight field, type or edit the technical complexity weighting. This indicates how much technical complexity you assign to a factor; for example, <i>the system is to be developed in ADA</i> might warrant a higher weight than <i>the system is to be a shell script</i> .	
5	In the Value field, type or edit a value representing the degree of influence the factor has on the project. As a suggested gauge: <ul style="list-style-type: none"> 0 indicates no influence 3 indicates average influence 5 indicates strong influence 	
6	Click on the Save button.	
7	Examine the Defined Technical Types list, and scroll across it to show the Ex Value column (Weight x Value). The summed Ex Values yield the Unadjusted TCF value (at the bottom of the dialog).	Environment Complexity Factors ^[518]

Step	Action	See also
	The Unadjusted TCF value is combined with the Environment Complexity Factors to skew the overall complexity up or down, depending on the level of technical complexity and the corresponding level of environmental support.	

Notes

- The TCF Weight evaluates its respective factor, but is irrelevant to a project; the **Value** field assesses each factor's role within a project and, for most purposes, is the only field requiring adjustment
- The supplied factors and their associated weights are defined by the *Use Case Points Method*, although they can be adjusted to suit a project's specific requirements
- You can transport the Technical Complexity Factors between models, using Export Reference Data and Import Reference Data

Learn more

- [Export Reference Data](#)^[306]
- [Import Reference Data](#)^[308]

3.6.8.2 Environment Complexity Factors

Environment Complexity Factors (ECFs) are used in the Use Case Metrics estimation technique.

The EABase.eap model contains a default set of ECFs, which you can add to or modify using the Estimation Factors dialog. This set of factors should include all factors that could affect the general design and development environment, including team experience and knowledge, team size, expertise and other non-functional environmental factors.

Access [Settings](#) | [Project Types](#) | [Estimation Factors](#) > [Environment Complexity Factors](#)

How to

To maintain Environment Complexity Factors

Step	Action	See also
1	On the Environment Complexity Factors tab, either: <ul style="list-style-type: none"> • Click on the New button to add another ECF, or • Click on the required factor in the Defined Environment Types list to edit it (go to step 3) 	
2	In the Factor Number field, type the appropriate ECF number.	

Step	Action	See also
3	In the Description field, type or edit the ECF description.	
4	In the Weight field, type or edit the environment complexity weighting. This indicates how much complexity you assign to a factor.	
5	In the Value field, type or edit a value representing the degree of influence the factor has on the project. As a suggested gauge: <ul style="list-style-type: none"> • 0 indicates no influence • 3 indicates average influence • 5 indicates strong influence 	
6	Click on the Save button.	
7	Examine the Defined Environment Types list, and scroll across it to show the Ex Value column (Weight x Value). The summed Ex Values yield the Unadjusted ECF value (at the bottom of the dialog). The Unadjusted ECF value is combined with the Technical Complexity Factors to skew the overall complexity up or down, depending on the level of technical complexity and the corresponding level of environmental support.	Technical Complexity Factors ^[517]

Notes

- The ECF Weight evaluates its respective factor, but is irrelevant to a project; the **Value** field assesses each factor's role within a project and, for most purposes, is the only field requiring adjustment
- The supplied factors and their associated weights are defined by the *Use Case Points Method*, although they can be adjusted to suit a project's specific requirements
- You can transport the Environment Complexity Factors between models, using Export Reference Data and Import Reference Data

Learn more

- [Export Reference Data](#) ^[306]
- [Import Reference Data](#) ^[308]

3.6.8.3 Default Hours

Setting an hourly rate is the most difficult factor in an accurate estimation. Typical ranges can vary from 10 to 30 hours per Use Case point.

Studying the *Use Case Points Method*, from which this variable is defined, can help you to understand its role in the estimation and facilitate selection of a suitable initial value.

The best way to estimate this value is through analysis of previous completed projects. By calculating the project estimation on a completed project for which the Use Cases and environment are configured within Enterprise Architect, you can adjust the hour rate to render an appropriate value for your unique work environment.

Access [Settings | Project Types | Estimation Factors > Default Hour Rate](#)

How to

To set the default hour rate per adjusted Use Case point

Step	Action	See also
1	In the Duration field, type the number of hours per Use Case Point.	
2	In the Hourly Rate field, type the cost per hour.	
3	Click on the Close button.	

Notes

- The values you enter are stored as local settings on your computer only
- This option is also active in the 'Lite', read-only version of Enterprise Architect

Learn more

- [Project Estimation Using Use Case Metrics](#)
- [The Read-only 'Lite' Edition](#) ^[26]

3.6.8.4 Estimating Project Size

Enterprise Architect uses a simple estimation technique based on the established:

- Number of Use Cases to be built
- Difficulty level of those Use Cases
- Project environment factors and
- Build parameters

This technique is of value only once you have developed a couple of known projects to use as a baseline. Please DO NOT use the provided 'guesstimates' as a real world measure until you have some real world base lines to measure against.

Access Select package in Project Browser:

Project | Use Case Metrics, or
Right-click | Documentation | Package Metrics

How to

On the Use Case Metrics dialog, work through the fields as indicated in the following table:

Field	Usage	See also
Root Package	Confirm the root package in the hierarchy. All Use Cases under this package could potentially be included in the report.	
Reload	Re-run the load from the selected package, usually after you change the filter criteria.	
Phase like	Include Use Cases with a phase that matches the wildcard value in the field. Use * to match any characters, for example 1.* for 1.1 and 1.2.	
Keyword like	Include Use Cases with a keyword that matches the wildcard value in the field. Use * to match any characters.	
Bookmarked	Include all Use Cases, or only those that are tagged, or those that are not tagged.	
Use Cases	Check the total count of Use Cases in the estimate. The Use Cases and their parameters are listed in the panel underneath this field.	
Include Actors	Select to include Actors in the estimate.	
Technical Complexity Factor	Review the parameters that describe the degree of technical complexity of the project. While the Unadjusted TCF Value comes from the Technical Complexity Factors tab of the Estimation Factors dialog, the other values compose the Use Case Points Method formula. Modify these fields with caution.	Technical Complexity Factors 517

Field	Usage	See also
	The final project estimate is directly proportional to the TCF.	
Environment Complexity Factor	<p>Review the parameters that calculate the degree of environmental complexity of the project, from factors such as programmer motivation or experience.</p> <p>The listed parameters compose the formula calculating the ECF, defined by the Use Case Points Method; the only parameter affected by the project is the Unadjusted ECF Value, derived from the Environment Complexity Factors ^[697] tab of the Estimation Factors dialog.</p> <p>The final project estimate is directly proportional to the ECF.</p>	Environment Complexity Factors ^[518]
Unadjusted Use Case Points (UUCP)	Check the sum of the Use Case complexity numbers.	
Ave Hours per Use Case	<p>Check the average number of hours assigned to easy, medium and difficult Use Cases.</p> <p>You cannot change these figures.</p>	
Total Estimate	<p>Review the detailed breakdown of the final figure.</p> <p>You must tailor the hours per Use Case Point figure to the level that matches your type of project and capability based on known previous project outcomes.</p>	
Default Rate	Set the default hours ^[520] to be fed into the final calculation.	Default hours ^[520]
Re-Calculate	Re-run the estimate, usually after you change the hours or Use Case point number.	
Report	Produce a rich text formatted report from the current estimate.	

3.6.9 SPEM

Topics

Topic	Detail	See also
Introduction	<p>According to the Object Management Group (OMG) Software & Systems Process Engineering Meta-Model Specification (Version 2.0, April 01 2008):</p> <ul style="list-style-type: none"> <i>The Software and Systems Process Engineering Meta-model (SPEM) is a process engineering meta-model as well as conceptual framework, which can provide the necessary concepts for modeling, documenting, presenting, managing, interchanging, and enacting development methods and processes. An implementation of this meta-model would be targeted at process engineers, project leads, project and program managers who are responsible for maintaining and implementing processes for their development organizations or individual projects.</i> <p>In 1999, the OMG placed a Request for Proposal concerning Software Process Engineering (SPE).</p> <p>In November 2002, the OMG released the Software Process Engineering Meta-model Specification 1.0.</p> <p>SPEM was defined as a Profile of UML, which used UML as a notation and took an object-oriented approach. To accommodate UML 2, the SPEM specification was upgraded to 2.0 in April 2008.</p>	<p>http://www.omg.org/spec/SPEM/2.0/PDF (Online Resource)</p>
SPEM in Enterprise Architect	<p>SPEM 2.0 focuses on providing the additional information structures that you require for processes modeled with UML 2 Activities or BPMN/BPDM, to describe an actual development process.</p> <p>Enterprise Architect enables you to develop SPEM diagrams quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer. The SPEM facilities are provided in the form of:</p> <ul style="list-style-type: none"> A SPEM diagram type, accessed through the New Diagram dialog A set of SPEM pages in the Toolbox, providing SPEM elements (stereotyped UML elements) SPEM element and relationship entries in the Toolbox Shortcut Menu and Quick Linker 	<p>Add New Diagram ^[72]</p> <p>SPEM Toolbox ^[52]</p> <p>Toolbox Shortcut Menu ^[70]</p> <p>Quick Linker ^[79]</p>

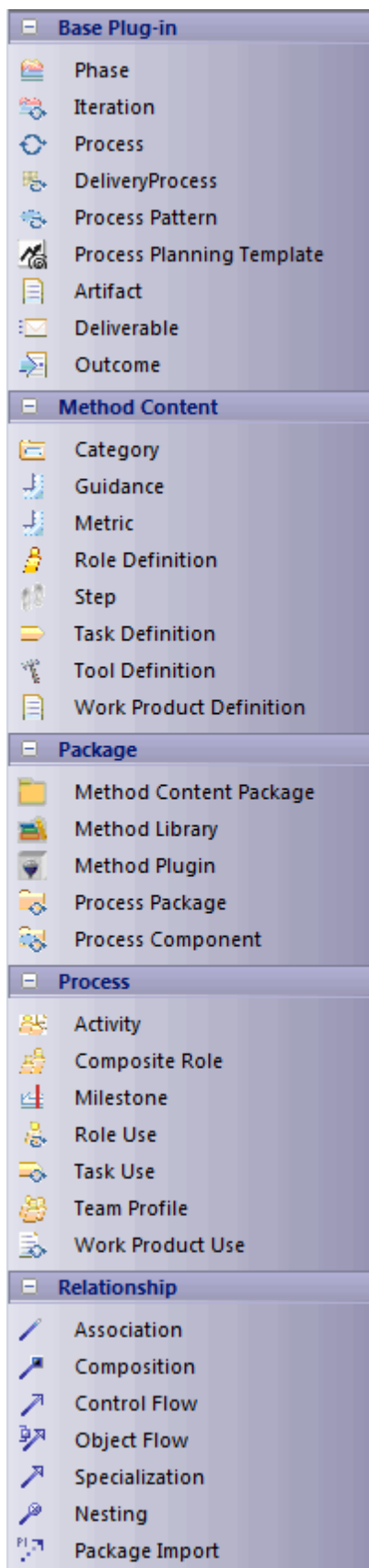
Topic	Detail	See also
Disable SPEM	If you prefer not to use SPEM in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).	MDG Technologies ^[1285]

3.6.9.1 SPEM Toolbox Pages

These pages provide the graphical SPEM elements for drawing the diagrams.

Access **View | Diagram Toolbox: More Tools | SPEM**

Example



Reference

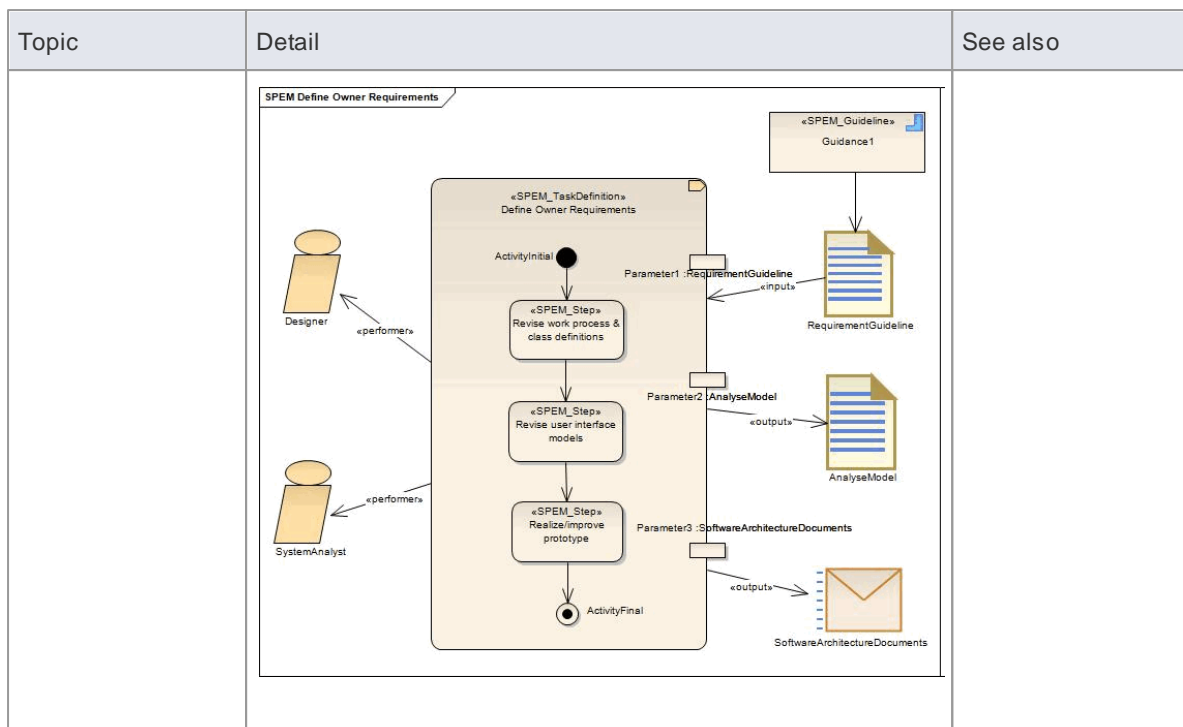
Page	Item	Action
Base Plug-in	Phase	Create a predefined special Activity representing a significant period in a project.
	Iteration	Group a set of nested Activities that are repeated more than once. Typically, Iteration is an Activity for which the default value of the <i>isRepeatable</i> attribute is True .
	Process	Represent a special Activity that describes a structure for particular types of development projects, or parts of them.
	Delivery Process	Represent a special Process describing a complete and integrated approach for implementing a specific project type.
	Process Pattern	Represent a special Process to describe a reusable cluster of Activities in a general process area that provides a consistent development approach to common problems.
	Process Planning Template	Represent a special Process that is prepared for instantiation by a project planning tool.
	Artifact	Represent a Work Product Definition that provides a description and definition for tangible work product types.
	Deliverable	Represent a Work Product Definition that provides a description and definition for packaging other Work Products, and that can be delivered to an internal or external party.
	Outcome	Represent a Work Product Definition that provides a description and definition for non-tangible work products.
Method Content	Category	Categorize content based on the user's criteria.
	Guidance	Identify reference items such as Guidelines, Templates, Checklists, Tool Mentors, Estimates, Supporting Materials, Reports and Concepts.
	Metric	Define a standard measurement for instances of Method Content elements.

Page	Item	Action
	Role Definition	Define a set of related skills, competencies, and responsibilities.
	Step	Represent parts or subunits of a Task Definition.
	Task Definition	Describe an assignable unit of work. Every Task Definition is assigned to specific Role Definitions. A Task is associated with input and output Work Products.
	Tool Definition	Describe the tools that are recommended or necessary for completing a specific Task.
	Work Product Definition	Define any forms of document, report or outcome that are consumed, produced or modified by Tasks.
Package	Method Content Package	Create a physical container to organize the Method Content elements.
	Method Library	Create an overall physical container for all SPEM 2.0 elements.
	Method Plugin	Create a physical container for Method Content Package and Process Packages. It can be used stand-alone as well as extended to many other Method Plugins.
	Process Package	Create a physical container that contains different kinds of Process element.
	Process Component	Create a special Process Package that provides the mechanism of encapsulation.
Process	Activity	Define basic units of work within a Process as well as the Process itself.
	Composite Role	Represent an aggregation of Role Definition references for an Activity.
	Milestone	Represent any significant events in a development project.
	Process	Create a special Activity that describes a structure for

Page	Item	Action
		particular types of development project.
	Role Use	Represent a Role Definition in the context of one specific Activity.
	Task Use	Represent a Task Definition in the context of one specific Activity.
	Team Profile	Define a nested hierarchy of teams and team members.
	Work Product Use	Represent a Work Product Definition in the context of one specific Activity.

Topics

Topic	Detail	See also
Presentation	<p>In Enterprise Architect, every SPEM stereotype can be presented in one of two ways:</p> <ul style="list-style-type: none"> • Iconic presentation, or • Textual presentation <p>The <i>iconstyle</i> tag is used for switching between these presentations. For example, in the SPEM diagram below, if you want the <i>SPEM_TaskDefinition</i> to have iconic presentation you set the <i>iconstyle</i> Tagged Value to True, and display the element as an icon.</p> <p>To get the textual presentation for <i>SPEM_TaskDefinition</i>, as an outline with a decoration in the top right corner, set the Tagged Value to False.</p>	



3.6.10 Update Package Status

Elements in Enterprise Architect can be assigned a current status, such as *Proposed*, *Validated* or *Mandatory*.

Often a complete package structure moves from one status to another (such as for release) in one operation. To help facilitate this, Enterprise Architect supports a 'bulk' update of package and element status, phase and version.

Access [Project Browser package context menu](#) | [Package Control](#) | [Update Package Status](#)

Use to

- Update the following element fields in the selected package:
 - Status
 - Phase
 - Version
- Define the scope of the update

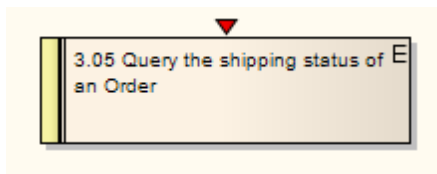
Reference

Field	Action	See also
New Status	Type in or select the new status of the package.	
New Phase	Type the new phase.	

Field	Action	See also
New Version	Type the new version.	
Modified Date	Defaults to today's date; if the change takes effect on a different date, use the 'spin' arrows to select that date.	
Set Date	Defaults to selected to apply the Modified Date; if necessary, deselect to ignore the date stamp on the change.	
Recursively update all child packages	Select to apply the changes through all child packages of the selected package; deselect to apply the changes to only the selected package.	
Include Elements	Defaults to selected; if the changes are to apply only to Package elements, deselect the checkbox.	
Include Element Requirements	Defaults to selected, to update the element requirements' Status field; deselect to ignore this field.	
Include Element Constraints	Defaults to selected, to update the constraints' Status field; deselect to ignore this field.	
OK	Click on this button to update all required elements to the new status.	

3.6.11 Manage Bookmarks

A bookmark is a visual clue that something is different about an element; it is represented by a small red triangle that displays above the element in the diagram.



You can assign any meaning to the bookmark as is appropriate to your model.

You can bookmark:

- Single elements manually, or
- All elements in a package automatically when they assume a defined status

Guide:

To work with bookmarks, select from the options described below:

Option	Usage	See also
Bookmark a selected element in a diagram	Either: <ul style="list-style-type: none"> • Press (Shift+Spacebar), or • Select the Edit Bookmark Selected menu option 	
Clear a bookmark on a selected element	Use either of the above two options again.	
Bookmark Multiple Elements in a Package	Select Project Browser package context menu Bookmarks The Manage Bookmarks dialog displays, enabling you to automatically bookmark any elements (and, if required, their children) in the package that have: <ul style="list-style-type: none"> • New changes defined in the Maintenance window • New defects defined in the Maintenance window, and/or • Test scripts defined in the Testing window This is useful to highlight elements that have additional project information.	
Clear all elements of bookmarks	To clear all bookmarks: <ul style="list-style-type: none"> • Throughout the current package, click on the Clear All button on the Manage Bookmarks dialog • In the current diagram, select the Edit Clear All Bookmarks menu option 	

Notes

- You should reload the project to update it with the new or cleared bookmarks
- The Model Search window enables searching based on bookmarked elements


Learn more

- [The Maintenance Workspace](#)^[223]
- [The Testing Workspace](#)^[221]
- [Refresh View of Shared Project](#)^[242]
- [Create & Modify Searches](#)^[62]

3.6.12 Monitor Change Events

You can automatically monitor work events in your project, using the Model Views facility.

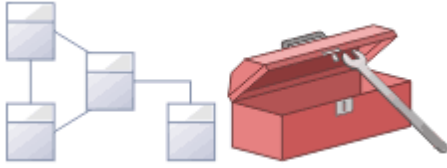
Guide:

Topic	Detail	See also
Model Views	<p>This facility enables you to:</p> <ul style="list-style-type: none"> • Automatically refresh a search in a View at an interval that you define • Notify you if the results of the search change between two consecutive searches <p>You can therefore use Model Views to monitor various events in the development project, depending on how you set up the search in a View.</p>	Model Views Operations 
Example	<p>You could set up a search to detect:</p> <ul style="list-style-type: none"> • Change items, or Issue items, so that Enterprise Architect would notify you as new items were created • Element Status, Type, Phase, Version, Priority and/or date of last update, so that Enterprise Architect would notify you as items were progressed to: <ul style="list-style-type: none"> • Fall in to the level of work represented by the search categories or • Move out of the categories into the next level of work • Tagged Values, so that - again - as items were changed to satisfy the criteria of a sequence of searches, the progression of items through a set of stages could be checked and managed 	
Personalization	<p>People responsible for different stages in a process could have their own Model View searches so that:</p> <ul style="list-style-type: none"> • As a development, validation or authorization task falls due the responsible person is automatically notified, and • When the work is complete both the next person in line and the overseeing manager are notified 	

Notes

- This facility is available in the Enterprise Architect Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions

3.7 Project Maintenance



This topic highlights some administrative functions you might have to carry out to maintain your model.

Topics

Topic	Detail	See also
Check the integrity of the data in a project	If you have a failed XML import, network crash or other unforeseen event that could disrupt the integrity of information in the model, it is recommended that you run a Project Integrity Check.	Check Project Data Integrity ^[535]
Reset Auto Increment Columns in Tables	XML Import/Export can affect the table auto increment column values and push them towards the maximum datatype value; you can re-sequence the columns to avoid this problem.	Reset Table Auto Increment or Identity Columns ^[536]
Upgrade an old project to enable use of new features	The structure of Enterprise Architect project files is occasionally changed to support more features; when this happens, existing project files must be upgraded to the new format to ensure correct operation and to take advantage of all the new features.	Upgrade Projects ^[537]
Rename a project	If you want to rename an Enterprise Architect project, you can do so through Windows Explorer.	Rename a Project ^[539]
Compact a project	After some time, a project .EAP file might benefit from compacting to conserve space.	Compact a Project ^[539]
Repair a project if it did not close properly	If a project has not been closed properly, such as during system or network outages, on rare occasions the .EAP file does not re-open correctly.	Repair a Project ^[540]

Notes

- You only rename, compact and repair models created as .EAP files; these processes are not required for models stored in a DBMS

3.7.1 Check Project Data Integrity

If you have a failed XML import, network crash or other unforeseen event that could disrupt the integrity of information in the model, it is recommended that you run the *Project Integrity Check* function.

You can select a variety of items to check. The integrity check examines all database records and ensures there are no 'orphaned' records or inaccurate or unset identifiers. This function does NOT check UML conformance, only the data relationships and repository structure.

You can run the integrity checker first in report mode to discover if anything should be corrected, and then run it again in 'recover/clean' mode.

When Enterprise Architect 'cleans' the model, it attempts to recover lost packages and elements, and generates a new package at the model root level called *_recovered_*. Check through any elements that are found and, if required, drag them into the model proper. If they are not required, delete them.

Access [Tools | Data Management | Project Integrity Check](#)

Use to

- Ensure that your project data is structurally complete

How to

To check the data integrity of your project

Step	Action	See also
1	Select the Project Integrity Check menu option. The Project Integrity Check dialog displays.	
2	Select the checkbox for each check to run: <ul style="list-style-type: none"> • Package Structure • Object Structure • Object Features • All GUIDs • Cross References • Connectors • UML 2.0 Migration (see below) 	
3	Select either: <ul style="list-style-type: none"> • the Report Only option to just view a report on the state of your model, or • the Recover/Clean option to recover and clean your project; before selecting this option back up your project file first 	
4	To write a log of the integrity check, click on the Save Results button and select a log	

Step	Action	See also
	file.	
5	<p>Click on the Go button to run the check.</p> <p>If you want to display the resulting information in a more readable layout, you can resize the dialog and its columns.</p>	

UML 2.0 Migration

The UML 2.0 Migration check enables you to migrate the project from UML 1.3 semantics to UML 2.0 semantics. The migration process currently converts activities that are invocations of operations into called operation actions as per the UML 2.0 specification.

The UML 2.0 Migration option is an exclusive process that does not enable any of the other checks to be selected. When you click on the **Go** button to perform the migration, a prompt displays for you to confirm the operation.

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Check Data Integrity** permission to perform a data integrity check

Learn more

- [List of Permissions](#) ^[264]
- [Run SQL Patches](#) ^[539]

3.7.2 **Reset Table Auto Increment or Identity Columns**

This topic explains possible impacts of XML Export/Import on table auto increment columns, and how to re-sequence the columns whose value approaches the maximum datatype value.

Topics

Topic	Detail	See also
XML Export/Import	<p>XML Export/Import can cause gaps in the numbering sequence of auto increment columns.</p> <p>Each XML Import deletes rows from several tables; the import then adds rows starting from the maximum value of the auto increment column.</p> <p>Repeated XML imports can result in the value of the auto increment approaching the maximum value of the database datatype; for example, SQL Server's <i>int</i> datatype has a maximum value of 2,147,483,647.</p>	

Topic	Detail	See also
Replication	<p>Large auto increment values can also arise where the project originated as an EAP replica or design master.</p> <p>The Jet engine assigns random values for auto increment columns with each XML Import into the project.</p> <p>These random values can approach the maximum range of the repository data type, which could present a problem when the EAP project is transferred to a repository.</p>	Replication ^[244]

Access [Tools](#) | [Data Management](#) | [Reset IDs](#)

How to

To re-sequence auto increment columns

Step	Action
1	Open the project.
2	<p>Select the Reset IDs menu option.</p> <p>A dialog displays listing all non-empty tables that contain an auto increment or identity column.</p> <ul style="list-style-type: none"> • The Rows column shows the number of rows in the table • The Maximum ID column shows the current maximum value of the auto increment column • The Action column shows either No Action or Reset, depending on how close the column value is to the datatype's maximum <p>Tables requiring a reset are automatically selected in the list.</p>
3	Click on the Go button to reset the auto increment column values.

3.7.3 Project Upgrade

The structure of Enterprise Architect project files is occasionally changed to support more features. In such cases, the project might have to be upgraded.

Topic	Detail	See also
Upgrading	<p>Upgrading to the new file structure is a simple and quick process that brings your project to the current level to:</p> <ul style="list-style-type: none"> • Ensure correct operation and • Support all the latest Enterprise Architect features 	

Topic	Detail	See also
Initial Check	When you load a project that was created in an early release of Enterprise Architect (for example, an archived project) using a recent release of Enterprise Architect, the system determines whether the project should be upgraded and, if the upgrade is necessary, displays the Upgrade Wizard.	
The Upgrade Project Wizard	The Upgrade Project Wizard takes you through the upgrade process and: <ul style="list-style-type: none"> • Advises you of the necessity to upgrade • Advises you to back up the current project; it is essential to back up before any changes are made • Checks which upgrade path is required • Guides you through the steps to perform the upgrade • Opens the newly converted project 	

Notes

- If you are using replication in your models, and the upgrade wizard detects that the project you are opening is a replica and not a Design Master, a different upgrade path is required
- Once upgraded, the project cannot be opened with the version of Enterprise Architect in which it was created

Learn more

- [Replication](#) ^[244]
- [Upgrade Replicas](#) ^[538]

3.7.3.1 Upgrade Replicas

Models that have replication features added might have to be upgraded differently from regular projects.

- If the model is a Design Master (the root model of all other replicas) then you can upgrade the model to suit the current version of Enterprise Architect; after upgrading a Design Master you should re-create the replicas, rather than synchronizing
- If the model is not a Design Master, Enterprise Architect must first remove the replication features, then upgrade the project in the normal manner; the Upgrade Wizard guides you through the steps

Learn more

- [Project Upgrade](#) ^[537]
- [Replication](#) ^[244]
- [Design Masters](#) ^[246]

3.7.4 Run SQL Patches

Occasionally, Sparx Systems might release a patch to correct a model fault.

Access [Tools](#) | [Data Management](#) | [Run Patch](#)

The patch generally checks how many records are to be updated, and reports on what is to be done.

3.7.5 Rename a Project

If you want to rename a project, you can only do so at the Windows file system level, using Windows Explorer.

How to

To rename an Enterprise Architect project .EAP file

Step	Action
1	If you have the project open, shut it down.
2	Ensure no other users have the file open.
3	Open Windows Explorer and navigate to the project.
4	Rename the project file using Windows Explorer.
5	You should keep the .EAP extension the same to preserve compatibility with the default project type, as installed in the registry at installation time.

3.7.6 Compact a Project

After a long period of use and development, a project .EAP file might benefit from compacting to conserve space.

Always compact and repair projects on a local drive, never on a network drive.

Access [Tools](#) | [Data Management](#) | [Manage .EAP File](#) | [Compact .EAP File](#)

How to

To compact a project

Step	Action
1	If you have the project open, shut it down.

Step	Action
2	Ensure no other users have the file open.
3	Select the Compact .EAP File menu option.
4	Follow the on-screen instructions to complete the process.

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Administer Database** permission to compact a project

Learn more

- [List of Permissions](#)^[264]

3.7.7 Repair a Project

If a project has not been closed properly, such as during system or network outages or on poor network connections, on rare occasions the .EAP file does not open correctly. A message displays informing you the project is of an unknown database format or is not a database file.

You can repair the project; however, never attempt to repair a project over a network connection; copy it to a local drive first.

Access [Tools | Data Management | Manage .EAP File | Repair .EAP File](#)

How to

To repair a project that was not closed properly

Step	Action	See also
1	Ensure that all users are logged off the project you are attempting to repair.	General ^[544]
2	Copy the project file to a local drive on your PC.	
3	In Enterprise Architect, select Tools Options to display the Options dialog. On the General page, deselect the Use Jet 4.0 - requires restart checkbox.	

Step	Action	See also
4	Close and restart Enterprise Architect and open a <i>copy</i> of the project you intend to repair as a <i>place holder</i> project, to enable access to the Repair .EAP File facility.	
5	Select the Repair .EAP File menu option, and follow the on-screen instructions.	
6	Once you have repaired the project, it is recommended that you perform a data integrity check.	Check Project Data Integrity ^[535]

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Administer Database** permission to repair a project

Learn more

- [List of Permissions](#)^[264]

3.8 Local Options



There are many options for customizing how Enterprise Architect displays and works with models and model elements. This topic describes those settings that are local to a particular user and machine.

Most of these settings are stored in your registry so they are set for your use only. For a networked workplace, registry settings can be copied down to any network workstation you log in to. Otherwise, the settings are valid for the current machine only.

Access **Tools | Options**

Select the required page of options by clicking on the appropriate category name in the left hand list on the dialog. For information on the options on a specific page, select the appropriate page title link below.

Topic	Detail	See also
General Options	Set options that define how Enterprise Architect operates as a whole.	General ^[544]
Standard colors of objects	Define the display and background colors of objects and properties across Enterprise Architect.	Standard Colors ^[546]
Diagram actions	Configure overall options for new diagrams and general diagram behavior.	Diagram ^[549]
Diagram appearance	Define how diagrams and their contents appear on display.	Diagram Appearance ^[550]
Diagram behavior	Define how a diagram responds to actions taken on it.	Diagram Behavior ^[553]
Options for Sequence diagrams	Configure various font settings and the focus of the control indicator for Sequence diagrams.	Diagram Sequence ^[555]
Object appearance	Configure how elements look and respond in diagrams.	Objects ^[556]
Link or Connector appearance	Define options for the creation, behavior and notation of connectors.	Links ^[559]
Colors of communication	Configure the colors used in Communication diagrams.	Communication Message Colors ^[561]

Topic	Detail	See also
messages		
XML Specification options	Configure various settings for working with XML.	XML Specifications ^[1862]
Source code engineering options	Set general options that apply to all languages when generating code from Enterprise Architect, including the default language in which to generate code.	Source Code Engineering ^[1892]
DDL and code editors	Configure options for Enterprise Architect's internal editor, and for the default editor for DDL scripts.	Code Editors ^[1895]
Constructor/ Destructor options	Configure options for Object Lifetimes.	Object Lifetimes ^[1898]
Attribute and Operation specifications	Configure Attribute and Operation specifications.	Attribute/Operations ^[1899]
Code language specifications	Define how Enterprise Architect handles a particular language when generating code, including the default source directory and the default editor for a specific code language.	ActionScript ^[1909]
		C ^[1911]
		C# ^[1912]
		C++ ^[1912]
		Delphi ^[1913]
		Java ^[1915]
		PHP ^[1916]
		Python ^[1917]
		Visual Basic ^[1920]
		VB.Net ^[1918]

Topic	Detail	See also

3.8.1 General

Access **Tools | Options > General**

Use to

- Set general options

Reference

Field	Usage	See also
Author	Set the default author when new elements are created and modifications made.	
Clipboard Format	Set the graphic format in which to save image to the clipboard; Metafile has the best detail.	
Double-click on Browser	Configure the Project Browser behavior.	Set Default (Project Browser) behavior ^[587]
Use Jet 4.0 - requires restart	<p>Set JET 4.0 as the database engine; this ensures compatibility with .EAP files that are in turn compatible with versions of MS Access later than Access 97, and that support unicode character sets.</p> <p>If your project is not in a Jet 4.0 database, you should also download a copy of the Jet 4.0 EABase Model from the Sparx Systems website, and do an EAP to EAP transfer of your model into the Jet 4.0 file.</p>	Perform a Project Data Transfer ^[442] Download link: Sparx Systems website
Use extended « and » characters	<p>Apply the guillemet characters to stereotypes.</p> <p>For some double byte character sets, it is best to select this checkbox.</p>	
Allow custom RTF tags in reports	<p>Enable you to use customized rich text format code in report templates when generating reports with the Legacy RTF Report Generator.</p> <p>From release 7.0 of Enterprise Architect, with the Notes formatting facility, this option is not really necessary.</p>	Legacy RTF Report generator ^[2318] Notes toolbar ^[979]
Project Directory	Specify the default location of Enterprise Architect projects.	

Field	Usage	See also
File Directory	Specify the default location for shared files defined on the Files tab of an element's Properties dialog (maximum 50 characters).	
Web Home	Specify the default home page to open when you click on the Home button in the internal web browser.	The Web Browser f1651
Web Search	Specify the default web page to open when clicking on the Web Search button in the internal web browser.	
Web Mail	Specify the email server address (http://xxxx/exchange/) for accessing email through the web browser within Enterprise Architect.	
Confirm Deletes	Use or bypass the Confirm Delete dialog; only clear this checkbox if you are an experienced user.	
Allow Free Sorting	Enable you to re-order elements within a package regardless of element type, in the Project Browser.	
Show Stereotypes	Show element and feature stereotypes in the Project Browser.	
Always open maximized	Ensure that Enterprise Architect always starts up in a maximized window.	
Hide Properties Info Section	Hide or show the properties information status bar on the Properties window.	
Docked Windows Tabs on Bottom (requires restart)	Display the docked window tabs at the bottom of the window (default). Clear the checkbox to show the tabs at the top of the windows.	
Main Diagram Tabs at Bottom (requires restart)	Display the diagram tabs at the bottom of the main view (default). Clear the checkbox to show the tabs at the top of the main view.	
File Loading Strategy	Select the Enterprise Architect model loading behavior for .EAP files; choose either: <ul style="list-style-type: none"> • Load on Demand (Lazy Load) • Preload Entire Model 	

Field	Usage	See also
	<p>Load on Demand does not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded.</p> <p>This means that a model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.</p> <p>For server connections, the Load on Demand option is set when the connection is created.</p>	

3.8.2 Standard Colors

The Standard Colors page of the Options dialog enables you to set the display color of a range of objects and their backgrounds. On first use, the page displays the system default colors. To display the range of colors available for an item, or to define a new color, click on the down arrow at the end of the appropriate field.

Access [Tools | Options > Standard Colors](#)

Use to

- Set global colors

Reference

Field	Usage	See also
Paper	Define the paper (background) color in diagrams.	
Element Fill	Define the fill color of elements.	
Element Line	Define the line color of elements.	
Shadow	Define the color of element outline shadows.	
Attribute	Define the color of attribute text.	
Method	Define the color of method (operation) text.	
Note Color	Define the note background color.	
Note Compartment Text	Define the color of text in the element Note compartment.	

Field	Usage	See also
Screen	Define the screen (element) color.	
Connector Line	Define the connector line color.	
Behavior	Define the color for behaviors in Activity diagrams.	
Custom Colors	Click on the Define Colors button to define up to 16 custom colors to be used in changing the appearance of elements on diagrams. The Show Project Custom Colors in Color Combo Boxes checkbox must be selected in order for this button to operate.	Create Project Colors ^[548]
Show Project Custom Colors in Color Combo Boxes	Enable the use of project custom colors within the Default Appearance dialog. When you select this option, the last 16 squares on the color selection palettes on the Default Appearance dialog are reserved for the custom colors. You must restart Enterprise Architect for changes to this option to take effect.	Set an Element's Default Appearance ^[817]

Notes

- Using this page of the Options dialog, you can set the background of all diagrams to be a specific color; you can also use the Diagram Appearance page to set all diagram backgrounds to be either a uniform color or to have a fade gradient from top to bottom
- As an alternative, you can create a background image for the diagram
- To override the default appearance of a specific element on all diagrams on which it is found, right-click on the element and select the **Appearance | Default Appearance** context menu option; the Default Appearance dialog displays
- To change the appearance of a specific element on the current diagram only, use the Format toolbar; if this toolbar is not displayed, select the **View | Toolbars | Format Tool** menu option

Learn more

- [Diagram Appearance](#) ^[550]
- [Create Custom Diagram Background](#) ^[757]
- [Default Appearance](#) ^[817]
- [Format Toolbar](#) ^[692]

3.8.2.1 Create Project Colors

This topic explains how you set the project colors to be made available in the element Default Appearance dialog.

Access [Tools | Options > Standard Colors: Define Colors](#)

Use to

- Set the project colors

How to

To define the custom colors for a project

Step	Action	See also
1	Click on the Define Colors button. The Color dialog displays.	
2	Click on the Define Custom Colors » button. The Color dialog is extended to provide color selection fields on the right of the dialog.	
3	Click on an empty block in the Custom Colors palette (or, if you are <i>replacing</i> a color, click on that color block in the palette).	
4	Create the required project color using the Luminosity slide bar, color mixer panel, Hue/Sat/Lum fields and/or Red/Green/Blue fields.	
5	Click on the Add to Custom Colors button to add the color to the selected Custom colors block.	
6	Repeat steps 3, 4, and 5 for any other project colors you want to define.	
7	Click on the OK button to close the Color dialog, then click on the Close button to close the Options dialog. The custom colors are made available for use in the last 16 squares on the color selection palettes on the Default Appearance dialog.	

3.8.3 Diagram

The New Diagram Defaults page of the Options dialog enables you to configure overall options for new diagrams and general diagram behavior.

Access [Tools | Options > Diagram](#)

Use to

- Configure diagram options and behaviours

Reference

Field	Usage	See also
Default Page Size	Show the default page size for new diagrams, which you set by clicking on the Page Setup button to display the Page Setup dialog.	
Print with Border	Print pages with a border.	
Landscape	Print pages in landscape orientation; this checkbox is controlled from the Page Setup dialog.	
Show Public Features Show Protected Features Show Private Features	Set the default visibility of Class features.	
Show Diagram Notes	Display the diagram details in the top left corner of all diagrams in the model; details include diagram name, package, version and author.	
Use Opaque Diagram Labels	Specify where opaque diagram labels should display; Screen and Printing are best, Clipboard and Files might not be desirable.	
Strict UML Syntax	Enforce compliance with UML syntax when adding new connectors and other structures. The syntax check is not performed for stereotyped elements that have had the <code>_metatype</code> attribute set.	Define a Stereotype as a Metatype ^[1312]
Disable fully scoped object names	Disable fully scoped object names, when an element is in a diagram; don't use when the element is in its home package. A scoped name is of the format <code>MyClasses::foo</code> , the <code>::</code> character indicating that the Class is within another	

Field	Usage	See also
	namespace.	
Allow change of Created Date	Enable the creation date on the Diagram Properties dialog to be altered.	
Zoom to best scale	Resize diagrams to neatly fit the screen.	
Auto-pan with middle mouse button	Turn on auto-panning using the middle mouse button; with this option off, the middle mouse button causes a different type of panning.	
Enable Connector line jumps	Defaults to selected, to enable connectors to show 'jumps' when they cross, on all diagrams in the model. Deselect to disable line jumps on all diagrams.	Connector Styles [948]
Scale Saved Bitmaps to	Enable Enterprise Architect to save bitmaps at a higher <i>resolution</i> , suitable for using in published works.	
Image Memory Limit	Set an image memory limit when generating images for RTF or HTML and when saving images to file. This is important when you have very large diagrams, as it affects the point at which Enterprise Architect starts to scale down the image; a low memory setting means it scales the image sooner.	
Diagram Frames	Select whether diagram frames are to be automatically added to images of diagrams in files saved to disk, print-outs, and the default Enterprise Architect clipboard. A diagram frame is a labeled outline around the diagram image, providing both a border and a reference.	Diagram Frame [1119]

3.8.3.1 Appearance

The Diagram Appearance page of the Options dialog enables you to define how diagrams and their contents appear on display.

Access [Tools | Options > Appearance](#)

Use to

- Configure the appearance settings of a diagram and elements

Reference

Field	Usage	See also
Configure Default Element Fonts	Set the default model and user text fonts.	Set default fonts ^[552]
Renderer	<p>Render smooth curves and diagonal lines in diagrams, so that staggered vertical or horizontal pixels are less obvious. Select the type of renderer you prefer to use:</p> <ul style="list-style-type: none"> • Basic is GDI32; it does not provide anti-aliasing and gradient fills • Enhanced-1 is parallel to Windows GDI+ but internal to Enterprise Architect; it provides anti-aliasing and gradient fills, and operates well across different platforms • Enhanced-2 is Windows GDI+; this can vary across different platforms, performing better than Enhanced-1 in some environments, and less well in others <p>Experiment with these options and see which works best for your system and requirements.</p>	
Default Element Border Width	Set the default element border width (in pixels).	
Scale view by	<p>Automatically increase the size of all objects on a diagram by up to 50%, without affecting other users reading that diagram.</p> <p>You can perform the same function with the Zoom Slider on the Status bar; changes in the 'zoomed' display scale of a diagram update this field and affect any other diagrams that you open.</p> <p>This has no impact any other diagram Zoom facility in Enterprise Architect.</p>	Status bar ^[152]
Print in Color	<p>Print your diagrams in color.</p> <p>Deselect the checkbox to print the diagrams in black and white.</p>	
Anti-aliased text	<p>Force text anti-aliasing in diagrams.</p> <p>If you deselect the checkbox, Enterprise Architect applies the MS Windows default setting. Therefore, if you do not want to use anti-aliasing, ensure that the Windows anti-aliasing default is also set to OFF.</p>	
Compress text to fit within Element	Determine the behavior of Enterprise Architect when text at <i>zoom</i> levels other than 100% would not fit inside the boundary of an element.	

Field	Usage	See also
	Enterprise Architect either compresses the text to fit within the boundary, or expands the element.	
Bold Element Names	Display element names in bold text.	
Element Shadows On	Display a shadow around the bottom and right edges of each element in a diagram.	
Invert rotated text for metafiles	Use different text format when external metafile readers are causing issues.	
Use Watermark	Add a watermark to any diagrams you print.	
Text	Define the watermark text, if a watermark is to be used.	
Show Gradient Fill for Paper Color	Switch between having a color gradient in the diagram background, or having a solid, uniform background color.	
Gradient Fill Direction For an Element	Select the direction for the color gradient within element boxes, or <none> for no color gradient.	

3.8.3.1.1 Set Default Fonts

Enterprise Architect enables you to define a standard font to apply across the model, or a font to apply to any diagrams you create personally.

You can define both, but the model font overrides any user font, to ensure that all members of a project team have a consistent and coherent view of the model. This avoids the problem of one user creating a diagram in a small font, and another user trying to view it in a larger font, which distorts the diagram.

It is recommended that a project authority sets the model default, and all project members abide by it and do not change it without project approval.

Access [Tools | Options > Appearance : Configure Default Fonts](#)

Use to

- Configure Font/s as the default for the model
- Configure Font/s as the default for diagrams created by the user

Reference

Field	Usage	See also
Model Font : Font Face	Specifies the font face used by the overall model.	
Model Font : Font Size	Specifies the font size used by the overall model.	
Clear	Clears the selections made for Font Face and Font Size.	
User Font: Font Face	Specifies the Font Face used on diagrams if no model font is specified.	
User Font: Font Size	Specifies the Font Size used on diagrams if no model font size is specified.	

Notes

- Both model and user fonts are overridden by specifically-defined element fonts, so that the element is viewed as designed regardless of the model or user defaults; to define the font for a specific element, right-click on the element in a diagram and select the **Appearance | Set Font** context menu option
- If you cannot read the diagrams because the default font makes the objects and text too small, you can scale up all objects (that is, all diagram displays) to a more readable size for you only; the objects are not scaled up for other users. Everything on the diagram is enlarged to the same extent, so it remains in proportion and readable. To do this, return to the Diagram Appearance page of the Options dialog and enter a suitable percentage value in the **Scale view by** field

Learn more

- [Set Element Font](#)^[839]
- [Appearance](#)^[550]

3.8.3.2 Behavior

The Diagram Behavior page of the Options dialog enables you to define how a diagram responds to actions taken on it.

Access **Tools | Options > Diagram : Behaviour**

Use to

- Define diagram behaviour

Reference

Field	Usage	See also
Auto Instance	Automatically create object instances when dragging certain element types - such as Class and Component - from the Project Browser, with the dragged element as the classifier.	
Instance has Classifier style	Automatically apply the classifier style of the element an instance is instantiated from when the instance is created.	
Show Life Lines	Show life lines for Sequence elements in non-Sequence diagrams.	
Layout uses all relations	Show all relationships in a diagram layout; deselect the checkbox to show only Generalizations and Associations.	
Auto Save Changes	Automatically save your changes as you work, without having to confirm prompts to do so.	
Show 'Hidden' Parents	Display any parents of elements in the diagram that are not part of the diagram.	
Auto Group Elements	Also move visually composed elements when moving diagram nodes; a node is considered composed if it is contained by the moved element and has a higher Z-order. Press and hold (Alt) whilst moving an element to toggle this option.	Diagram Context Menu ^[687]
Show Linked Items in Package	Display connected items on packages.	
Show Package Item Scope	Display the + and - indicators representing the scope of the items.	
Shift-Mouse Autoscroll	Enable you to press and hold (Shift) and use the mouse to autoscroll around diagrams.	
Use Automatic SubActivities	Generate a new Structured Activity linked to the diagram from a Structured Activity diagram dragged from the Project Browser.	
Always Highlight Context Element	Show a hatch border around a selected element.	Highlight Context Element ^[822]

Field	Usage	See also
Objects Snap to Grid	Snap all elements to the grid lines.	
Show Grid	Display the grid.	
Grid Size	Specify the grid size, if you have selected Objects Snap to Grid .	
Auto Tidy	Automatically tidy line angles for custom connectors; this 'nudges' the custom line into horizontal and vertical increments.	Connector Styles [95]
Tidy line gap	Specify the amount Enterprise Architect should enable you to move a line away from horizontal and vertical when you are tidying lines for custom connectors. (See Auto Tidy above).	Connector Styles [95]
Alias only	Display the alias instead of the element name on elements with aliases.	
Alias and Name	Display both the element name and the Alias in the format <i>(Alias) name</i> .	
Use alias field for role	Replace the Alias property of instances with a Role property.	

3.8.3.3 Sequence

The Sequence page of the Options dialog enables you to configure various font settings and the focus of the control indicator for Sequence diagrams.

Access [Tools | Options > Diagram : Sequence](#)

Use to

- Configure font settings
- Configure the focus of the control indicator for Sequence diagrams

Reference

Field	Usage	See also
Message Spacing	Specify the vertical gap (in points) between Sequence messages (can be overridden manually by dragging a message up or down).	
Default page layout is Landscape	Set the default orientation of Sequence diagrams to landscape.	
Show Sequence Numbering	Show sequence numbers on Sequence messages.	
Assume right to left messages are Return	Automatically generate return messages.	
Default concurrency is Asynchronous	Set the default concurrency for Sequence Messages to Asynchronous ; deselect to set the default concurrency to Synchronous .	Message (Sequence Diagram) ^[1232]
Width	Select the line width (in points) of the 'focus of control' rectangle (thick part of lifeline).	
Assume message returns	Assume implicit returns when none are explicitly drawn (recommended).	
GarbageCollect	Automatically truncate lifelines for created elements after the last message (that is, assume garbage collect rather than explicit delete).	
Name	Display the MS Windows Font dialog, click on (...) and define the font of the caption bar heading (above your diagram); this is particularly useful for non-English character sets.	
Size	Specify the size of the heading font (this overrides the font size in the Font dialog, above).	
Use System	Apply the Enterprise Architect system default heading font.	

3.8.4 Objects

The Objects page of the Options dialog enables you to configure how elements look and respond in diagrams.


Access [Tools | Options > Objects](#)

Use to

- Change how elements look and respond

Reference

Field	Usage	See also
Version	Set the default version for new elements.	
Phase	Set the default phase for new elements.	
Highlight References	Highlight parameters in operations that are passed by reference rather than value.	
Reference Char(s)	Specify a character to use for the reference.	
Prefix/Suffix	Indicate whether to use the Reference Char(s) value as a prefix (before) or a suffix (after).	
Warn about spaces in class names	Enable or hide the warning message that a Class, operation or attribute name has embedded spaces (which can cause coding problems).	
Classes honor analysis stereotypes	Show Classes as their stereotype; for example, if a Class is stereotyped as a Boundary, it appears as a Boundary rather than a Class.	
Show stereotype icon for requirements	Show or hide a code letter in the top right corner of Requirement (E , for external), Change (C) and Issue (I) elements.	
Support for Composite Objects	Enable you to drag child elements onto parent elements in a diagram, and automatically embed them (and drag embedded child elements out of parent elements, breaking the child-parent relationship).	Move Elements Within Diagrams [805]
Auto-resize marks diagram 'dirty'	Ensure that auto-resizing of elements (such as Classes) marks the current diagram as changed (asterisk on the diagram name tab), so it should be saved.	
Highlight {abstract} elements	Highlight abstract elements with a suitable tag <i>{abstract}</i> in the top right of the Class.	

Field	Usage	See also
Allow elongated Use Cases	Stretch Use Cases or Use Case extension points with long names to enable space for the name. If you deselect the checkbox, Use Case re-sizing is proportional and the 'make same' menu options are blocked.	Position Submenus [112]
Show status colors on diagrams	Enable color coding for Requirements and similar elements.	Status Types [995]
Copy inheritance links on duplicate	Duplicate Inheritance and Realization connectors when an edit/copy is performed.	
Port and Part type visible by default	Enable Port and Part types to be shown by default.	
Show buttons for selected Object on diagram	Display the floating toolbar buttons to the left of the selected object 	
Edit Object on New	Automatically show the element Properties dialog when a new element is added.	
Show «column» stereotype	Hide or show the « <i>column</i> » stereotype used when data modeling.	
Extend Complexity	Extend levels of complexity to five levels in the Complexity option in the Properties window; otherwise only three levels are available.	
UML 1.5 Components	Use UML 1.5 components (Enterprise Architect versions 4.0 and later support UML 2.x).	
Show State Compartment	Show or hide the State Compartment divider under the state name.	
Show Duplicate Tags	Enable duplicate tags to be shown.	
Group Operations by Stereotype	Group an element's operations by their stereotype on the diagram.	

Field	Usage	See also
Group Attributes by Stereotype	Group an element's attributes by their stereotype on the diagram.	
Sort Features Alphabetically	Sort element features alphabetically. Features include Attributes, Operations, Tags, Constraints and Test Cases.	
Disable spelling	Turn off automatic spell checking. Deselect to resume automatic spell checking.	Spell Checking ^[480]
Advanced	Set the visibility of certain elements in reports and in diagram packages.	Element Visibility ^[559]

3.8.4.1 Element Visibility

Some elements do not appear in packages and in RTF output by default. Click on the **Advanced** button on the Objects page of the Options dialog to specify which elements should be visible.

Access [Tools | Options > Objects : Advanced](#)

Use to

- Specify which elements should be visible within RTF reports

Learn more

- [Objects](#) ^[556]
- [Customize Visibility of Elements](#) ^[813]

3.8.5 Links

The Links page of the Options dialog provides options for the creation, behavior and notation for connectors.

Access: [Tools | Options > Links](#)

Use to

- Provides options for connector behaviour

Reference

Field	Usage	See also
Edit Connector on New	Automatically show the connector Properties dialog when a new connector is added.	
Association default = source --> target	Set the direction of new Associations to source->target (that is, with an arrow head at the target).	
Generalization link style Default = Tree	Show Generalizations as tree style hierarchies.	
Shade Qualifier boxes	Lightly shade all Qualifier boxes.	
Draw Aggregations Reversed	Draw Aggregate and Composite connectors from target element to source element. When deselected (the default), these connectors are drawn from source to target.	
Prompt on connector deletes	Display a prompt before deleting connectors, offering the choice of hiding the connector on the diagram or deleting it completely. If you deselect this option, the delete operation defaults to the last setting on the dialog.	Delete Connectors [954]
Suppress Link Constraints	Suppress connector constraints in diagrams.	
Suppress Qualifier boxes	Suppress boxes when displaying qualifiers.	
Show Uses arrowheads	Show an arrowhead on Actor->Use Case Associations.	
Show Override Operation dialog on new connector	Show the Override Operation dialog automatically when adding generalizations and realizations between Classes and Interfaces, if the target element has features that can be overridden.	
Suppress ' + ' Role Scope	Ensure that the role and scope are not displayed on the diagram.	
Pen Width	Set the default connector width.	

Field	Usage	See also
Routing	Set the default connector style for new connectors.	
Enable	Enable the Quick Linker.	The Quick Linker ^[79]
Show Help	Add a 'help' menu option to the end of the Quick Linker menu.	
Center to center	Change the position of the dashed guide line for new connectors.	
Exact placement		
Force perpendicular line		

Notes

- For **Draw Aggregations Reversed**, all tools have the parent as the target and the child as the source of the connector, that is a requirement of UML; only the direction in which you drag the mouse to draw the connector is changed

3.8.6 Communication Message Colors

The Communication Message Coloring page in the Options dialog enables you to configure the colors used in Communication diagrams.

Access **Tools | Options > Communication Colors**

Use to

- Configure colors used in Communication diagrams

Notes

- Select the **Use Communication Message Coloring** checkbox to turn on colored messages. When you enable this option, Communication messages appear in different colors depending on the sequence group they belong to on a diagram; for example, 1.n are black, 2.n are red, 3.n are green
- Click on the down arrow in each color field, and click on the appropriate color for the message group. Set the color sequence as required; the pattern repeats after 8 sequence groups

3.8.7 XML Specifications

The XML Specifications page of the Options dialog enables you to configure various settings for working with XML.

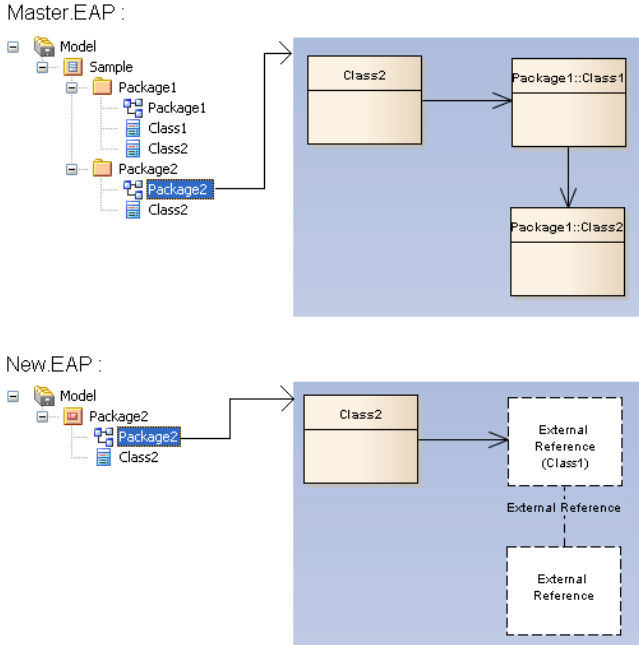
Access [Tools | Options > XML Specifications](#)

Use to

- Configure various XML Settings

Reference

Field	Usage	See also
Editor	Set the default editor for any XML documents you open within Enterprise Architect.	
Format XMI Output	Set whether or not formatting is applied to your XML output.	
Write Log	Set whether or not to write to a log file when you import or export XML.	
Use DTD	Set whether or not to use a Data Type Definition.	
Export Diagrams	Set whether or not to export diagrams when you export XML.	
Export Alternate Images	Set whether or not to export the alternative images used in the model when you export to XML.	Using the Image Manager ^[755] Check in a Package ^[365] Sharing Reference Data ^[305]
Code Page	<p>Set the Code Page to use.</p> <p>Setting a NULL encoding string results in the encoding tag being entirely omitted from the XML output.</p> <p>Click on the Default button to restore the setting to the default Code Page.</p>	
Export Diagram Images	Set whether or not to export diagrams as images when you export XML.	

Field	Usage	See also
Image Type	Define the format of the image to export to if Export Diagram Images is selected.	
Create placeholders for missing External References during XMI 1.1/2.1 Import	<p>Set this option to show, during an Enterprise Architect exported XML 1.1/2.1 file import, any missing:</p> <ul style="list-style-type: none"> • Diagram elements, using a Boundary element as a placeholder • Diagram connectors, using a Dependency connector as a placeholder <p>When importing an Enterprise Architect exported XML 1.1/2.1 file with cross-package references, use this option to visually show the diagram items that are not yet in the model.</p> <p>A sample scenario would be exporting <i>Package2</i> to XML 1.1 from <i>Master.EAP</i> and importing it into <i>New.EAP</i> - the Classes <i>Class1</i> and <i>Class2</i> and the connector between them are shown using a placeholder on the diagram <i>Package2</i>.</p>  <p>The diagram illustrates the import of Package2 into New.EAP. The top part shows Master.EAP with Package1 containing Class1 and Class2, and Package2 containing Class2. The bottom part shows New.EAP where Package2 contains Class2, but references to Class1 and the connector are shown as dashed boxes labeled 'External Reference'.</p>	
Prefix EA Tagged Values in XMI 1.0 with ea\$	Set whether or not to prefix any Enterprise Architect Tagged Values within any XML 1.0 you create, with ea\$.	
Update Last Save Time	Set whether to update the timestamp of the last time controlled packages were saved.	

Field	Usage	See also
Report Cross Package Reference Deletions (XMI 1.1)	Set this option to report any package cross-references that would be deleted when a Controlled Package is exported to XMI 1.1.	Report Deletion of Cross-Package References ^[433]
Default XML Directory	Define the default XML directory to use when importing and exporting XML.	

Notes

- If **Export Alternate Images** is set, and you have packages in your model under version control, then any alternative images used in those packages are also exported to the version control repository when you check in the packages; in this case, you would only select the checkbox if the alternative images are subject to frequent change

Otherwise, do not select this option and instead use **Export/Import Reference Data** to manage alternative images

Learn more

- [Export MOF to XMI](#) ^[2097]

Part



4 Navigate, Search and Trace



This section explains how you **navigate** through the model structures in Enterprise Architect, locate and display specific information, and trace the origins, development and fate of the model elements. While the **Project Browser** provides an effective hierarchical view of the model, large models can present particular challenges, and tracing **dependencies** and **relationships** is not always a simple task. Fortunately, Enterprise Architect has a number of tools designed to meet just these requirements.

Topics

Topic	Link
The use of the Project Browser and Package Browser to navigate and explore your model	The Project Browser ^[567] Package Browser ^[587]
How Model Views provide different perspectives and 'entry points' into a model	Model Views ^[598] Diagram Slide Show ^[607]
Finding elements and relationships using the Model Search tools and custom queries	Model Search ^[611] Pre-defined Searches ^[617]
Traceability tools to help track completeness, dependencies and other behavioral and structural relationships	Traceability Tools ^[636]
Use of the Diagram List or Package Browser to view model elements in a conventional list based manner	Diagram List ^[596] Package Browser ^[587]
Use of the Relationship Matrix to understand and track relationships between model elements in a spreadsheet view	Relationship Matrix ^[639]
Using Diagram Filters to highlight information in diagrams	Diagram Filters ^[629]
Using the Traceability window to explore model dependencies	The Traceability Window ^[637]
Finding and working with Element Relationships	The Relationships Window ^[648]

4.1 Navigate: Exploring Your Model



Navigating is the process of systematically exploring the structure of your model using the hierarchical and list based views, such as the Project Browser. It is particularly useful when you are familiar with the structure of a model and its packaged contents. Whereas Searching provides a flat and random set of results, the Project and Package Browsers provide a highly structured viewpoint, reflecting the exact structure of your model.

Topics

Topic	Link
Use the Project Browser, the main interface element for exploring and navigating your model	The Project Browser ^[567] Project Browser Context Menus ^[569] Project Browser Toolbar ^[583]
The Package Browser - a tabular, editable view of elements in a selected package, which can be displayed in the main workspace	Package Browser ^[587]
The Diagram List - a tabular, editable view of elements in a selected diagram, which can be displayed in the main workspace	Diagram List ^[596]
Use Model Views to provide different perspectives and 'entry points' into your model	Model Views ^[598]
Set up a Diagram Slide show	Diagram Slide Show ^[607]
Use the Pan and Zoom window to navigate around very large diagrams	The Pan & Zoom Window ^[610]

4.1.1 The Project Browser

The Project Browser enables you to navigate through the Enterprise Architect project space. It displays packages, diagrams, elements and element features in a tree like structure, reflecting the arrangement of elements and packages within your model. The Project Browser is the primary mechanism for browsing and exploring your model and is the jumping off point for many of the most important features in Enterprise Architect.

Access **View | Project Browser (Alt + 0)****Use to**

- Navigate your model
- Drill down to specific elements
- Drag and drop elements and packages within the model
- Copy (duplicate) whole packages
- Import/export model information
- Set up version control, locking, execution analysis and many other aspects of the model tied to a particular package
- Document models
- Import code, xmi and csv files, database schema and other external data sources

If you right-click on an item in the Project Browser to display the context menus, you can perform additional actions such as adding new packages, creating diagrams, renaming items, creating documentation and other reports, and deleting model elements. You can also edit the name of any item in the Project Browser by selecting the item and pressing (**F2**).

Views

The Project Browser can be divided into Views, each of which contains diagrams, packages and other elements. A default View hierarchy is described below, but you can create different Views to suit your requirements:

View	Description
Use Case View	The functional and early analysis View, containing Business Process and Use Case models.
Dynamic View	The dynamics of your system, containing State Charts, Activity and Interaction diagrams.
Logical View	The Class Model and Domain Model View.
Component View	A View for your system components. The high level view of what software is to be built (such as executables, DLLs and components).
Deployment View	The physical model; what hardware is to be deployed and what software is to run on it.
Custom View	A work area for other Views, such as formal requirements, recycle bin, interview notes and non-functional requirements.

View	Description

Selective Collapse:

When you are working on an expanded project in the Project Browser, you might want to locate the parent element or package of an item, and/or collapse the structure under that parent element or package.

How to

To focus on the parent element or package

Step	Action	See also
1	Position the cursor on an item within the element or package.	
2	Press (←) on the keyboard to highlight the parent.	
3	Press the key again to collapse the structure under that parent element or package.	

Learn more

- [Views](#) ^[676]
- [Project Browser Icon Overlays](#) ^[585]

4.1.1.1 Project Browser Context Menus

This section describes the many options available to work on objects in the Project Browser, selected from the following context menus:

- [Model \(Root Node\) context menu](#) ^[569]
- [Package Context Menu](#) ^[571]
- [Element Context Menu](#) ^[571]
- [Diagram Context Menu](#) ^[581]
- [Operation Context Menu](#) ^[582]

4.1.1.1.1 Model (Root Node) Context Menu

The *Root Node* in the Project Browser is the *Model* element. You can have more than one model element.

The first level packages beneath the Model node are sometimes referred to as *Views* as they commonly divide a model into categories such as Use Case Model and Logical Model.

Right-click on the Root Node to display the **Model** context menu.

Reference

Action	Usage	Shortcut	See also
Add-In	Access the facilities of each Add-In currently enabled for the project.		
Scripts	List the scripts enabled for execution directly from the Project Browser. (This does not display if no Project Browser scripts exist.)		
Package Control	Display the Package Control submenu.		Package Control ^[424]
Rename Model	Rename the current model.		
Add	Display the Add submenu; see below.		
Copy Package to Clipboard	Copy the selected package to the clipboard, to be copied into another package in the same .eap file or a different .eap file.		Copy the selected package ^[683]
Paste Package from Clipboard	Paste a package from the clipboard into the selected package.		Paste a package ^[683]
Find in Project Browser	Find a specified term in the Project Browser.	Ctrl + Shift + F	
Expand Branch	Expand all items.		
Collapse Branch	Collapse all items.		
Import Model from XMI	Import a model from an XMI file.	Ctrl + Alt + I	Import a model ^[41]
Export Model to XMI	Export a model to XMI.	Ctrl + Alt + E	Export a model ^[41]
Rich Text Format (RTF) Report	Produce RTF documentation for the model.	F8	Produce RTF documentation ^[225]
HTML Report	Produce HTML documentation for the model.	Shift + F8	Produce HTML

Action	Usage	Shortcut	See also
			documentation <small>[2352]</small>
Diagrams Only Report	Produce a Diagrams Only report (in RTF) for the model.	Ctrl + Shift + F8	Diagrams Only Report <small>[2340]</small>
Copy Reference	Copy a reference to the root node to the Enterprise Architect clipboard. Select the appropriate sub-option to copy the: <ul style="list-style-type: none"> selected package hierarchy structure (node path) or node GUID 		
Delete Project Root	Delete the Model node and all subordinate Views and packages.		
Help	Display the Help topic for the Project Browser.		

Add Submenu:

Action	Usage	Shortcut	See also
Add Model (root node)	Create a new model root, without any subordinate structure or content.		
Add View	Create a new View (package).		
Add a New Model using Wizard	Add further models using the Model Wizard. This option enables you to	Ctrl + Shift + M	Model Wizard <small>[660]</small>

4.1.1.1.2 Package Menu

Right-click on a View or Package in the Project Browser. The context menu displays, providing the following options:

Action	Usage	Shortcut	See also
Add-In	Access the facilities of each Add-In currently enabled for the project.		

Action	Usage	Shortcut	See also
Scripts	List the scripts enabled for execution directly from the Project Browser. (This option does not display if no Project Browser scripts exist.)		Script Group Properties ^[2375]
Properties	Display the Properties dialog for the Package element.		Properties Dialog ^[843]
Package Control	Submit packages to package control and version control.		Controlled Package Menu ^[424] Package Version Control Menu ^[357]
Add	Add a new diagram, element or another package to the current package.		Add sub menu ^[574]
Paste Diagram	If you have copied a diagram from another package, paste the diagram into the currently-selected package.		
Lock Package	Set a lock on the package.		Lock Package ^[273]
Package Browser	Display the Package Browser, showing the elements contained in the selected package.		Package Browser ^[587]
Find in All Diagrams	List all diagrams in which this package is depicted, in the Element Usage dialog.	Ctrl + U	Show Element Use ^[802]
Turn On Level Numbering (Turn Off Level Numbering)	Add a sequence number to each element in the package, based on the element's position in the package hierarchy. For nested elements, the numbering indicates level; that is: 3.2 3.2.1 3.2.1.1 This option is only available for packages, and the numbering only applies to the elements in the package, not diagrams. If elements are added, moved or deleted from the package, the numbering automatically		Model Requirements ^[1484]

Action	Usage	Shortcut	See also
	adjusts.		
Linked Document	Create or display a linked document for the package or view.	Ctrl + Alt + D	Linked documents ^[928]
Delete Linked Document	Delete the linked document attached to the package. The system prompts you to confirm the deletion.		
Documentation	Produce a variety of reports and documentation in RTF format.		Documentation Sub-menu ^[575]
Code Engineering	Perform Code Engineering functions.		Code Engineering Sub-menu ^[575]
Execution Analyzer	Display the Execution Analyzer window and highlight the Analyzer Scripts for the selected Package. If no Analyzer Scripts have been configured you are given the opportunity to create one.		Visual Execution Analyzer ^[2143]
Import/Export	Import and export using XML text files.		Import/Export Sub-menu ^[577]
Transform Current Package	Perform a model transformation on the selected package.	Ctrl + Shift + H	Transform Elements ^[1683]
Contents	Reorganize the package contents after making changes.		Contents Sub-menu ^[577]
Bookmarks	Bookmark all elements in the selected folder.		Manage Bookmarks ^[530]
Find in Project Browser	Search the Project Browser for specific elements (identified using the Find in Project Browser dialog).	Ctrl + Shift + F	
Copy Reference	Copy a reference to the package to the Enterprise Architect clipboard; select the appropriate sub-option to copy the: <ul style="list-style-type: none"> • Selected package hierarchy structure (node path) or • Node GUID 		

Action	Usage	Shortcut	See also
Copy Package to Clipboard	Copy the selected package to the clipboard, to be copied into another package in the same .eap file or a different .eap file.		Copy a Package ^[683]
Paste Package from Clipboard	Paste a package from the clipboard into the selected package.		Copy a Package ^[683]
Paste Element(s) from Clipboard	Paste elements copied to the clipboard into the selected package.		Copy Elements Between Packages ^[808]
Save Package as UML Profile	Save the selected package as a Profile.		UML Profiles ^[1277]
Set View Icon	Change the display icon for the selected package (View level).		
Move up	Move the package up in the list.		
Move down	Move the package down the list.		
Delete <packagename>	Delete the selected package and its contents.		
Help	Display the Help topic for the Project Browser.		

4.1.1.2.1 Add Sub-Menu

In the Project Browser, right-click on a package and select the **Add** context menu option.

Action	Shortcut	See also
Add Diagram		New Diagram ^[724]
Add Element	Ctrl + M	New Element ^[798]
Add Package	Ctrl + W	Add Package ^[681]
Add a New Model using Wizard		Model Wizard ^[660]

4.1.1.1.2.2 Documentation Sub-Menu

In the Project Browser, right-click on a package and select the **Documentation** context menu option.

Action	Shortcut	See also
Rich Text Format (RTF) Report	F8	RTF Document ^[2251]
HTML Report	Shift + F8	HTML Report ^[2353]
Diagrams Only Report	Ctrl + Shift + F8	Diagrams Only Report ^[2340]
Testing Report		Test Document ^[2230]
Open in Relationship Matrix		Element Relationship Matrix ^[639]
Generated Report Options		RTF Report Options ^[2260]
Copy RTF Bookmark		Copy RTF Bookmark ^[2346]
Package Metrics		Package Metrics ^[520]

4.1.1.1.2.3 Code Engineering Sub-Menu

In the Project Browser, right-click on a package and select the **Code Engineering** context menu option.

Action	Description	Shortcut	See also
Generate Source Code		Ctrl + Alt + K	Generate Source Code ^[1768]
Import Source Directory		Ctrl + Shift + U	Import Source Directory ^[1795]
Import Binary Module			Import Binary ^[1797]
Synchronize Package With Code	Reverse engineer or forward engineer the model and code to synchronize them and incorporate changes made in one but not in the other. You can also update individual Class	Ctrl + Alt + M	Update Package Contents ^[1774] Element Menu - Project Browser ^[577]

Action	Description	Shortcut	See also
	elements from the source code, either through the Project Browser or on a diagram.		Operation Menu - Project Browser ^[582] Code Engineering Menu Section ^[836]
Generate DDL			Generate DDL ^[1979]
Import DB schema from ODBC			Import DB schema from ODBC ^[1973]
Generate XML Schema			Generate XSD ^[2049]
Import XML Schema			Import XSD ^[2053]
Generate WSDL			Generate WSDL ^[2085]
Import WSDL			Import WSDL ^[2086]
Reset Options for this Package			Reset Options ^[1922]
Reset DBMS Options			Reset DBMS Options ^[1945]
Set as Namespace Root Clear Namespace Root			Namespaces ^[1777]
Live Code Generation	Toggle between using and not using live code generation; that is, to update your source code instantly as you make changes to your model.		

4.1.1.1.2.4 Execution Analyzer Sub-Menu

In the Project Browser, right-click on a package and select the [Execution Analyzer](#) ^[2143] context menu option. This menu option displays the Execution Analyzer window and highlights the Analyzer Scripts for the selected Package. If no any Analyzer Scripts have been configured you are given the opportunity to create one.

4.1.1.1.2.5 Import/Export Sub-Menu

In the Project Browser, right-click on a package and select the **Import/Export** context menu option.

Action	Description	Shortcut	See also
Import package from XMI file		Ctrl + Alt + I	Import package [417]
Export package to XMI file		Ctrl + Alt + E	Export package [413]
CSV Import / Export			CSV Import [440] / Export [439]

4.1.1.1.2.6 Contents Sub-Menu

In the Project Browser, right-click on a package and select the **Contents** context menu option.

Menu Option	Action	See also
Expand Branch	Expand all of the items in the Project Browser.	
Collapse Branch	Collapse all of the items in the Project Browser.	
Reset Sort Order	Return sorting of package contents to list in alphabetical order.	
Reload current package	Refresh the current package in the Project Browser.	Refresh Model View [242]

4.1.1.1.3 Element Menu - Project Browser

Right-click on an *element* (such as Class, Object, Activity or State) in the Project Browser to display the element's context menu.

Command	Action	Shortcut	See also
Add-In	Access the facilities of each Add-In currently enabled for the project.		
Scripts	List the scripts enabled for execution directly from the Project Browser. (This option does not display if no Project Browser scripts exist.)		Scripts [2375]
Properties	View and modify the element properties.		

Command	Action	Shortcut	See also
Add	Create a child element and diagram (Classifier elements) or a connector to another element.		Add sub menu ^[580]
Paste Diagram	Pastes a copy of a diagram from the clipboard as a child diagram of the selected element. The Copy Diagram dialog displays, offering the choice between a shallow copy and a deep copy.		Copy (Duplicate) Diagram ^[742]
Rule Composer	For a Rule Task element, invoke the Rule Composer tab in Business Rule Modeling.		Rule Composer ^[1536] tab
Attributes	Display the Attribute dialog ready to create a new attribute.		
Operations	Display the Operations dialog ready to create a new operation.		
Create Workbench Instance	Create workbench variables from the selected Class. When you select this option, Enterprise Architect prompts you to name the variable, which then displays in the Workbench window.	Ctrl + Shift + J	Create workbench variables ^[2177] Workbench ^[2176] window
Generate Code	Generate the source code for this element.	F11	Generate Source Code ^[1768]
Synchronize with Code	Synchronize the element in the Project Browser with the source code. You can also synchronize Class elements from the source code on a diagram, and reverse engineer or forward engineer model packages and code to synchronize them and incorporate changes made in one but not in the other.	F7	Import Source Code ^[1790] Operation Menu - Project Browser ^[582] Update Package Contents ^[1774] Code Engineering Menu Section ^[836]
View Source Code	View the source code of the element.	F12	Editing Source Code ^[1798]
Open Source Directory	Open the source directory containing the source code for the element.	Ctrl + Alt + Y	

Command	Action	Shortcut	See also
Find In All Diagrams	Locate the element in all open diagrams. This option also operates on Port and Part Property Type Classifiers.	Ctrl + U	
Locate in Current Diagram	Select the element in the currently-visible diagram. If the element is not in the diagram, this option is grayed-out.		
Copy RTF Bookmark	Copy a bookmark in RTF format to the clipboard.		Bookmark ^[2346]
Linked Document	Create a Linked Document (Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions).	Ctrl + Alt + D	Linked Documents ^[928]
Delete Linked Document	Delete the linked document attached to the selected element.		
Add Custom Reference	Set up cross references between elements in a diagram and the selected element in the Project Browser.	Ctrl + J	Cross references ^[803]
Copy Element(s) to Clipboard	Copy the selected element or elements to the clipboard to be pasted into another package in this .eap file or another .eap file.		Copy the selected element ^[808]
Copy Reference	Copy a reference to the element to the Enterprise Architect clipboard. Select the appropriate sub-option to copy the: <ul style="list-style-type: none"> selected element hierarchy structure (node path) or node GUID 		
Move Up	Move the element up in the list of elements within this package.		
Move Down	Move the element down in the list of elements within this package.		
Delete '<element Name>'	Delete the element.		

Command	Action	Shortcut	See also
Help	Display the Help topic for the Project Browser.		

4.1.1.1.3.1 Add Sub Menu

Access Right-click element in Project Browser: (Insert)
 Right-click element | Add

Menu Option	Action	See also
Activity	Add an Activity behavior element and one of its associated diagrams to the selected Classifier element.	Activity ^[110] Object Classifiers ^[89]
Interaction	Add an Interaction behavior element and one of its associated diagrams to the selected Classifier element.	Interaction Occurrence ^[113] Object Classifiers ^[89]
State Machine	Add a State Machine Behavior element and one of its associated diagrams to the selected Classifier element.	State Machine ^[116] Object Classifiers ^[89]
RuleFlow Activity	Add a Rule Flow Activity element and associated diagram to a Class, in Business Rule Modeling.	Create a Rule Flow Model ^[153]
Add Diagram	Create a diagram to explain or expand on the selected Classifier element, using the New Diagram dialog.	Add New Diagrams ^[72]
Create Link	Create a connector to another element.	Create Connector in Project Browser ^[95]

Notes

- Elements such as Actors, Classes and Activities can define a large amount of information that can be conveniently represented by or expanded in a child diagram; the **Add** sub-menu for these elements provides all of the options listed above
- Elements such as Timing, Exit and History have much more specific functions that do not require expansion; the **Add** sub-menu for these elements only provides the **Create Link** option

4.1.1.1.4 Diagram Menu - Project Browser

Right-click on a diagram in the Project Browser to open the **Diagram** context menu.

Action	Usage	Shortcut	See also
Extensions	Access the facilities of each Add-In currently enabled for the project.		
Scripts	List the scripts enabled for execution directly from the Project Browser. (This option does not display if no Project Browser scripts exist.)		Scripts ^[2375]
Properties	View and modify a diagram's properties.	F5	
Open	Open the diagram in the Diagram View.		
View Diagram As	Offers options to display the diagram contents as: <ul style="list-style-type: none"> • The Diagram List, listing the elements in the selected diagram • A Gantt chart 		Diagram List ^[596]
Copy Diagram	Copy the diagram for pasting into another location.		Copy a Diagram ^[742]
Copy RTF Bookmark	Copy a bookmark in RTF format to the clipboard.		Bookmark ^[2346]
Add Custom Reference	Add this diagram as a cross reference to other elements.		Cross reference ^[803]
Print Diagram(s)	Print the currently-selected diagram or diagrams (hold (Ctrl) or (Shift) while selecting).		
Copy Reference	Copy a reference to the diagram node to the Enterprise Architect clipboard. Select the appropriate sub-option to copy the: <ul style="list-style-type: none"> • selected hierarchy structure (node path) or • node GUID 		

Action	Usage	Shortcut	See also
Compare to Baseline	Displays the Manage Baselines dialog, from which you select a Baseline of the parent package to review and - if necessary - reverse any differences between the diagram and baseline.		Manage Baselines ^[398] Check Visual Changes to Diagrams ^[403]
Move up	Move the diagram up in the list of diagrams within this package.		
Move down	Move the diagram down in the list of diagrams within this package.		
Delete '<diagram name>'	Delete the selected diagram.		
Delete selected items	Delete several selected diagrams (hold (Ctrl) or (Shift) while selecting).		
Help	Display the Help topic for the Project Browser.		

4.1.1.1.5 Operation Menu - Project Browser

To display the **Operation** (or Method) context menu, right-click on an Operation in the Project Browser.

Action	Usage	Shortcut	See also
Add-In	Access the facilities of each Add-In currently enabled for the project.		
Scripts	List the scripts enabled for execution directly from the Project Browser. (This option does not display if no Project Browser scripts exist.)		Script Group Properties ^[2375]
Generate Code	Generate code for the operation.	F11	
Synchronize With Code	Synchronize the operation with the code. You can also synchronize Class elements from the source code on a diagram, and reverse engineer or forward engineer model packages and code to synchronize them and incorporate changes made in one but not in the other.	F7	Import Source Code ^[1790] Element Menu - Project Browser ^[577] Update Package Contents ^[1774]

Action	Usage	Shortcut	See also
			Code Engineering Menu Section ⁸³⁶
View Source Code	Open the Source Code Viewer and display the operation.	F12	Editing Source Code ¹⁷⁹⁹
Operation Properties	Display the Properties dialog for the operation.		
Copy Reference	Copy a reference to the operation to the Enterprise Architect clipboard. Select the appropriate sub-option to copy the: <ul style="list-style-type: none"> Selected hierarchy structure (node path) Node GUID 		
Delete Operation	Delete the operation.		
Help	Display the Help topic for the Project Browser.		

Notes

- You can display an equivalent context menu for an attribute by right-clicking on the attribute in the Project Browser

4.1.1.2 Project Browser Toolbar

The Project Browser toolbar enables you to perform a range of operations on your project structure.

Access **View | Project Browser**

Use to

- Perform a variety of operations on the project structure and components

Reference









Action	Usage	Shortcut	See also
New Model	Create a new Model Package in the project,		Model Wizard ⁶⁶⁰

from Pattern	from a predefined UML or Technology pattern.		
Add a Package	Create a new child package under the selected package.		Add a Package ^[68]
New Diagram	Create a new child diagram under the selected package or element.		Add New Diagrams ^[72]
Create Element	Create a new child element under the selected package or element.		Add Elements Directly to Packages ^[79]
Find in Project Browser	Perform a simple search for a text string in the Project Browser.		
Documentation	Provide options to generate, on the selected package in the Project Browser, an: <ul style="list-style-type: none"> • RTF report • HTML report or • Diagram Only report 	(F8) (Shift+F8) (Ctrl+Shift+F8)	Generate RTF Documents ^[225] HTML Reports ^[235] Diagrams Only Report ^[234]
Code and DB Generation	Provide options to: <ul style="list-style-type: none"> • Generate source code • Generate DDL • Import a source directory • Import a binary module • Import a database schema • Generate package contents to synchronize with package code • Reset the source code language <p>All these options operate on the selected package.</p>		Generate a Package ^[177] Generate DDL for a Package ^[197] import a Directory Structure ^[179] Import Binary Module ^[179] Import Database Schema ^[197] Update Package Contents ^[177] Reset Options ^[192]
Move Element Up	Move the selected package or element further up the Project Browser, within its parent package.		
Move Element Down	Move the selected package or element further down the Project Browser, within its parent package.		





Help	Open the Enterprise Architect Help on the Project Browser.		
-------------	--	--	--

4.1.1.3 Project Browser Icon Overlays

The Project Browser displays the status of each package in the model by overlaying status icons on the package icon. The following table describes what each overlaid icon means.

Icon Overlay	Indicates that...	See also
	This package is controlled and is represented by an XML file on disk; version control either is not being used or is not available. You can edit the package.	Controlled Packages ^[422]
	This package is version controlled and checked out to you, therefore you can edit the package.	Use Version Control ^[353]
	This package is version controlled and not checked out to you, therefore you cannot edit the package (unless you check the package out).	Use Version Control ^[353]
	This package is version controlled, but you checked it out whilst not connected to the version control server. You can edit the package but there could be version conflicts when you check the package in again.	Offline Version Control ^[322]
	This package is a namespace root, which denotes where the namespace structure starts; packages below this point are generated as namespaces to code.	Namespaces ^[1777]
	The version-controlled package has been flagged as <i>Read Only</i> , using a package control method applied through the Automation Interface - as a script, for example. The icon displays with other package control icons except for the 'blue box' icon (above) which takes precedence. The <i>Read Only</i> flag can be applied to packages whether they are checked-in or checked-out.	Package Class ^[2433]
<MDG Add-In icon>	MDG Add-Ins specify their own icon to denote that this branch of the model belongs to that Add-In. All packages connected to an MDG Add-In correspond to a namespace root, so the namespace root icon is not displayed.	MDG Add-Ins ^[2671]

Similarly, the Project Browser indicates attribute and operation scope status with icons. The following table describes what each indicator icon means.

Icon Overlay	Indicates that...	See also
 	The attribute or operation is scoped as protected.	
 	The attribute or operation is scoped as private.	

Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions, if project User Security is on, the Project Browser also has element locking indicators (red and blue exclamation marks) that indicate the lock status of individual elements and packages

The availability of these elements for editing depends on whether user locks are required or not

Learn more

- [Configure User Security](#)^[25]
- [Locked Element Indicators](#)^[27]

4.1.1.4 Order Package Contents

Ordering elements is very important when it comes to structuring your model, especially packages. RTF documents honor any custom ordering when printing documentation. Enterprise Architect enables you to change the order of elements listed in the Project Browser.

By default, elements are first listed in:

- Order of type, then
- Order of set position, then
- Alphabetically

You can use the context menu options to move an element up or down within its type, but not outside its type. This means you can re-sequence Packages or Diagrams or Use Cases, but you cannot mix up elements.

However, you can change this default behavior to allow elements to be re-ordered within the package, regardless of type.

Learn more

- [Set Default Behavior](#)^[58]

4.1.1.5 Set Default Behavior

The General page of the Options dialog provides several options for altering the look and behavior of the Project Browser.

Access [Tools | Options > General](#)

Panel	Option	Effect	See also
Double-click On Browser	Shows Properties	When an item in the Project Browser is double-clicked, the item's Properties dialog (if available) displays.	
	Opens Branch	When an item in the Project Browser is double-clicked, the hierarchy expands to show the item's children. If there are no children, nothing happens.	
	Opens Branch & Diagram	As above, but also opens the first diagram beneath the item, if applicable.	
Project Browser	Allow Free Sorting	Enables you to re-order elements within a package regardless of type, within the Project Browser.	Order Package Contents ^[586]
	Show Stereotypes	When a stereotype is defined for an element, the stereotype name then displays in front of the element name. You set the stereotype of an element in its Properties dialog. Shut down and restart Enterprise Architect to enable this change to take effect.	

4.1.2 Package Browser


The Package Browser is a tabular, editable view of elements in a selected package, which can be displayed in the main workspace.

Access [Project Browser > package | View | Package Browser \(Ctrl + Alt + R\)](#)
[Project Browser > Right-click package | Package Browser Diagram > Right-click package | Package Browser](#)

Use to

- Streamline the process of creating and updating elements in a package selected from the Project Browser; this can be particularly useful for analysts to create and maintain formal requirement definitions within the model
- Print the list or generate an RTF document directly from the entries on the list

Topics

Topic	Detail	See also
The Package Browser	The Package Browser is related to the Diagram List, which lists the elements in a selected diagram; however, the Package Browser has a wider range of options.	Diagram List ^[590]
Formats	<p>The Package Browser has two formats:</p> <ul style="list-style-type: none"> • <i>User Defined</i> format, where you can change how the element information is structured on the screen using the value-grouping band below the toolbar • <i>Model Hierarchy</i> format, where the package and element hierarchies are represented in the display <p>You switch between these formats using the Show Hierarchy button () in the toolbar.</p>	List Header ^[590]
Change display	<ul style="list-style-type: none"> • Sort the items by any column value in ascending or descending order, by clicking on the column header; initially the elements are listed in numerical order (if level numbering is turned on in the Project Browser) or alphabetical order within type • Apply a diagram filter for elements (the Fade, Gray Scale and Hide modes reduce the number of elements listed; Select mode has no effect) • Change the sequence of columns, by dragging column headers left or right • In <i>Model Hierarchy</i> format, the Name column is always on the left; you cannot move any other column into that position, although you can rearrange the order of the rest of the columns <p>Because of this, if you group or sort information in <i>User Defined</i> format and switch to <i>Model Hierarchy</i> format your information structure is altered, and it is not restored when you switch back to <i>User Defined</i> format</p>	Diagram filters ^[629]
Display Element Properties	Double-click on the element entry to display the element Properties dialog.	

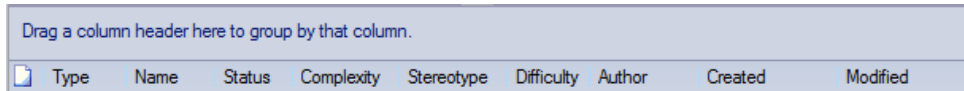
Topic	Detail	See also
Show Elements in a Child Package	Double-click on the child package in the list to display the names of the child elements.	
Select Items	Select: <ul style="list-style-type: none"> • an element by clicking on it • a specific value by clicking twice on it (not double-clicking); either the value becomes directly editable or the Properties dialog displays in which you can edit the value • several individual elements by holding (Ctrl) as you click on them • a range of elements by holding (Shift) as you click on the first and last in the range 	
Add new items to the package	Click on a listed element and press (Ctrl+N) or (Insert), or right-click and select the Add New context menu option <ul style="list-style-type: none"> • in <i>Model Hierarchy</i> format, new elements are inserted in the order in which they appear in the Project Browser • in <i>User Defined</i> format, new items are inserted to comply with any sort order and/or grouping; if the list is not sorted or grouped, the items are added to the end 	
Add a child element to the selected element (<i>Model Hierarchy</i> format)	Press (Esc) and then (Ctrl+N); otherwise, elements are added as siblings of the selected element. You can add child elements only when the whole row is selected, with none of the row cells or fields highlighted; press (Esc) to remove selection from an individual cell.	
Add elements to the Package Browser	Drag the required elements from anywhere in the Project Browser. You can also drag elements <i>from</i> the Package Browser <i>into</i> the Project Browser.	
Delete elements from the list	Select the item and press (Ctrl+D). In <i>Model Hierarchy</i> format, you cannot delete a parent element until all its child elements have been removed or deleted.	
Include each element's notes (documentation)	Notes are shown underneath the element. To add or edit notes, click on the item and press (Ctrl+Space) to transfer control to the Notes window, in which you create or edit the note text.	Package Browser Options ^[592]


Topic	Detail	See also
Review the element's custom (advanced) properties	Click on the item and press (Ctrl+Enter). The Custom Properties dialog for the element displays.	
Further Work	You can do further work on the Package Browser using the toolbar and context menu options.	Package Browser Options ⁵⁹²

4.1.2.1 List Header

This topic describes the organization of the column headings and contents for various reports in Enterprise Architect.

Topics

Topic	Description
View Header	 <p>The View Header defines the columns of information that are presented by the report, and the order in which data items are presented.</p> <p>To review the column headings:</p> <ul style="list-style-type: none"> Right-click on the header to display the Field Chooser context menu option; the Field Chooser dialog displays <p>This enables you to add columns to or remove columns from the output; between them, the View Header and Field Chooser dialog show the full range of column headers available.</p> <p>To add a column heading to the View Header:</p> <ul style="list-style-type: none"> Drag it from the Field Chooser dialog onto the header to the position you want the column of data to display When you have selected the column headings you require, click on the red cross in the top right corner of the Field Chooser dialog to close it <p>To remove a column from the output:</p> <ul style="list-style-type: none"> Drag the column heading downwards out of the View Header <p>To change the sequence of columns:</p>

Topic	Description
	<ul style="list-style-type: none"> • Drag the required column header left or right, as appropriate
Value Grouping	<p>For most reports, and when the Package Browser or Diagram List is in <i>User Defined</i> format, you can organize the reported data according to the value of one or more of the column categories (this facility is not available in the Package Browser in <i>Model Hierarchy</i> format).</p> <p>You might organize the data by Type, and within each Type by Name; if you then click on any of the other column headings, the data within this grouping is further sorted with the values of the selected column (for example, Created) in ascending or descending order.</p> <p>To set up the value grouping, drag the column heading representing the primary grouping (such as Author) onto the Drag a column header here to group by that column field.</p> <p>Now drag the column heading for the next level of grouping (such as Status) to the right of the first heading; the two heading titles display as connected blocks, as shown below:</p>  <p>You can, if required, add further levels of grouping by dragging other column headings onto the hierarchy (such as Created), and restructure the order by dragging existing or additional headings into the level you want them to hold.</p> <p>For example, you could make Type the secondary grouping by dragging it to the right of Author, or drop Status between Author and Type.</p> <p>To remove a grouping level, drag the appropriate column heading out of the sequence and below the View header; any subordinate groupings move up a level.</p>
Filter Bar	<p>The Filter Bar displays underneath the column headings; you can hide or show it using the Toggle Filter Bar icon in the toolbar.</p> <p>The Filter Bar consists of a row of fields, one field per column; as you type a value into one of these fields, the Package Browser results are automatically filtered to show entries with a value in the corresponding column that match the value in the field. The filter is case sensitive.</p> <p>For example, if you type the letter F in the Filter Bar field under the Author column, the results immediately reduce to entries for which the author name includes Fred, Frank, Felix, Felicity, Fairmile, Farmer or Fitzgibbon (but none where the name includes f only as a lower case letter, such as Rufus).</p> <p>If you then go on to type Fi, the results immediately reduce again to only entries for which the author name includes Fitzgibbon.</p>

Learn more









- [Package Browser](#)⁵⁸⁷
- [Diagram List](#)⁵⁹⁶
- [Report View](#)⁴⁷⁷





4.1.2.2 Package Browser Options

The Package Browser Toolbar and right click Context menu provide a wide range of tools and functions necessary for building and managing models. A few examples include adding new packages and elements, adding new diagrams, generating reports, software engineering, MDA transformations, version control and model export/import.

The Toolbar and Context menu are two powerful ways to access much of Enterprise Architect's primary modeling toolset.

Toolbar:

Icon	Action	Shortcut	Link
	Add a new element to the package.	Ctrl + N	Add Elements Directly to Packages <small>[798]</small>
	Display the Notes window, to add or edit notes for the selected element.	Ctrl + Space	
	Delete the selected element.	Ctrl + D	
	Print the current contents of the Package Browser.		
	Display the Generate RTF Documentation dialog, to create one RTF report on all selected element (s).		Generate RTF Documentation Dialog <small>[2256]</small>
	<p>Select:</p> <ul style="list-style-type: none"> The appropriate element type from the drop-down list, or All to list all objects <p>The report then lists only elements of that specific type.</p>		
	Select a UML, Extended or MDG Toolbox category to specify the category of elements shown in the filter list (above).		
	<p>Toggle between:</p> <ul style="list-style-type: none"> Including child packages and their contents in the list, and Showing only the first-level contents of the 		

	selected diagram or package		
	<p>Toggle the display between:</p> <ul style="list-style-type: none"> Model Hierarchy format and User-Defined, value-grouping format 		List Header ^[590]
	<p>Display a short menu from which you can select to:</p> <ul style="list-style-type: none"> Hide the contents of the Notes compartment of each element Display the first few words, or Display the full text 		
	Change the display to show the contents of the parent package of the selected package.		
	Display Help on the Package Browser.		

Context Menu Options:

Action	Usage	Shortcut	See also
Properties	Display the Properties dialog for the selected element.		
Edit Notes	Add or edit notes on the element, in the Notes window.	Ctrl + Space	
Add New	<p>If the Filter List field in the toolbar is set to:</p> <ul style="list-style-type: none"> All, display the New Element dialog, through which you create an element of the required type A specific element type, add an element of that type to the package or diagram in the Package Browser and the Project Browser 		
Find in Diagrams	<p>Display:</p> <ul style="list-style-type: none"> The diagram that uses the element or, 		Show Element Use ^[802]

Action	Usage	Shortcut	See also
	<ul style="list-style-type: none"> If the element is used in multiple diagrams, display the Element Usage dialog, which lists the diagrams that contain the element 		
Find In Project Browser	Highlight the selected element in the Project Browser.		
Bookmark Item	Bookmark the element.		
Create Linked Document (Edit Linked Document)	Create (or edit) a Linked Document. (Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions.)	Ctrl + Alt + D	Linked Documents ^[928] Link Document to UML Element ^[929] Edit Linked Documents ^[930]
Delete Linked Document	(If the element has a linked document.) Delete an existing linked document.		
Documentation	<p>Generate an RTF report; you have two options:</p> <ul style="list-style-type: none"> Generate a separate report on each selected object in the report Generate one report on all selected objects. <p>In either case, the Generate RTF Documentation dialog displays.</p> <p>You also have the option to print out the Package Browser list itself.</p>		Generate RTF Documentation Dialog ^[2256]
Sort Contents	<p>In <i>Model Hierarchy</i> format, synchronize the list with the Project Browser hierarchy, to ensure that all element and package hierarchies and sequences are - if necessary - updated.</p> <p>Normally changes are updated automatically, but there can be delays if changes are made outside the Package Browser.</p>		
Reload	Reload the list of elements to refresh the order and content with any recent changes.		

Action	Usage	Shortcut	See also
Add Tag Value Column	Add a column to show the values for a selected Tagged Value, for each element in the list (see below).		
Print	Print the Package Browser.		
Delete Selected	Delete the selected element from the Package Browser.		

Add Tagged Value Column:

If you want to show the value that an element has for a particular tag (and the value that any other element in the list has for that tag), you can add a *read-only* column for the tag. There are two ways in which to do this:

Method	Action	See also
1	Open the Tagged Values window for the element and drag the tag onto the Package Browser; a new column is added for each tag you drag onto the list.	
2	<p>Right-click on the element in the Package Browser or Diagram List and select the Add Tag Value Column context menu option; the Add Tag Value Column dialog displays.</p> <p>Either type the tag name into the Tagged Value field, or select one of the two options:</p> <ul style="list-style-type: none"> • Global Tagged Values - populates the Tagged Value drop-down list with all global Tagged Values in the project; this enables you to select a Tagged Value that is applicable to a large number of elements • Tagged Values from the selected Element - populates the Tagged Value drop-down list with the Tagged Values defined for the selected element (as listed on the Tagged Values window for that element); this enables you to filter the initial selection to a specific set of tags <p>When you have selected the required tag, click on the OK button to insert a column for that tag into the Package Browser or Diagram List header.</p>	

Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if Auditing is turned on and the Package Browser is open, you can view a history of changes to any selected element or connector, in the Audit History tab of the Output window (if Security is enabled, you must have at least **Audit View** permissions to display the audit history)

Learn more

- [Auditing](#)^[382]
- [Audit History](#)^[392]
- [List of Permissions](#)^[264]

4.1.3 Diagram List

The Diagram List is a tabular, editable view of elements in a selected diagram, which can be displayed in the main workspace.




Access **diagram | Diagram | Switch To | List View**
Project Browser > Right-click diagram | View Diagram As | List View
Project Browser > diagram (Ctrl+Alt+R)
diagram > Right-click | List View

Use to

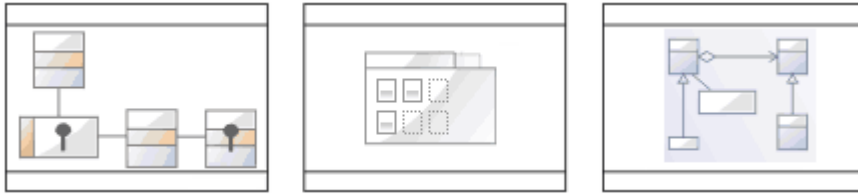
- Streamline the process of creating and updating elements in a diagram selected from the Project Browser; this can be particularly useful for analysts to create and maintain formal requirement definitions within the model
- Print the list or generate an RTF document directly from the entries on the list

Topics

Topic	Detail	See also
The Diagram List	The Diagram List is related to the Package Browser, which lists the elements in a selected package; however, the Package Browser has a wider range of options.	Package Browser ^[587]
Formats	The Diagram List is displayed in <i>User Defined</i> format, where you can change how the element information is structured on the screen using the value-grouping band below the toolbar.	List Header ^[590]
Change display	<ul style="list-style-type: none"> • Sort the items by any column value in ascending or descending order, by clicking on the column header; initially the elements are listed in numerical order (if level numbering is turned on in the Project Browser) or alphabetical order within type • Apply a diagram filter for elements (the Fade, Gray Scale and Hide modes reduce the number of elements listed; Select mode has no effect) • Change the sequence of columns, by dragging column headers left or right 	Diagram filters ^[629]
Display Element Properties	Double-click on the element entry.	

Topic	Detail	See also
Select Items	Select: <ul style="list-style-type: none"> • an element by clicking on it • a specific value by clicking twice on it (not double-clicking); either the value becomes directly editable or the Properties dialog displays in which you can edit the value • several individual elements by holding (Ctrl) as you click on them • a range of elements by holding (Shift) as you click on the first and last in the range 	
Add new items to the diagram	Create new elements on the Diagram List, using the toolbar or context menu options; these elements are automatically added to the diagram.	Package Browser Options [592]
Delete elements from the list	Select the item and press (Ctrl+D).	
Include each element's notes (documentation)	Notes are shown underneath the element. To add or edit notes, click on the item and press (Ctrl+Space) to transfer control to the Notes window, in which you create or edit the note text.	Package Browser Options [592]
Review the element's custom (advanced) properties	Right-click on the item and select the Properties context menu option, then select the Advanced page.	
Further Options	You can do further work on the Diagram List using the same toolbar and context menu options as for the Package Browser, except that the Diagram List <i>does not have</i> : <ul style="list-style-type: none"> • The <i>Model Hierarchy</i> format and hence does not have the Show Hierarchy button () in the toolbar • The  toggle button • The  toggle button The Diagram List <i>does have</i> these additional context menu options: <ul style="list-style-type: none"> • Diagram View - show the elements of the selected diagram as a diagram instead of as the Diagram List • Gantt View - show the list as a Gantt chart of resource allocations • Diagram Properties - display the Diagram Properties dialog for the diagram 	Package Browser Options [592] Diagram Properties [725]

4.1.4 Model Views













The Model Views window enables you to encapsulate your model into the areas you are interested in.

Access [View | Model Views](#)

Topics

Topic	Detail	See also
View Root Nodes	<p>There are three types of View root-node available:</p> <ul style="list-style-type: none"> • <i>Model Views</i> - stored in the model and visible to all users; you can have many of these • <i>My Views</i> - stored locally on your machine and visible only to you; you can have only one of these • <i>Technology-defined Views</i> - read only; each View is stored with and populated by the corresponding active MDG Technology <p>Additionally, there is a <i>Recent Discussions</i> folder that contains current correspondence from the Team Review concerning items that are held in any of the Views; this folder has a separate repository of postings for each team review server connection you access through the model. You can control how recent these postings must be, and how many are to be listed.</p> <p>When you open the Model Views window for the first time on a project, a <i>Model Views</i> root section, <i>My Views</i> root section and <i>Recent Discussions</i> folder are added for you. These can not be deleted or renamed; however, you can create further Model View root nodes that you can modify and delete.</p>	<p>Model Views Operations ⁶⁰⁴</p> <p>Team Review Tools ²⁷⁹</p> <p>Team Review Connections ²⁹⁶</p> <p>Model Views Context Menus ⁶⁰⁷</p>
Subordinate Folders	<p>Under the <i>My Views</i> root node you can add a single level of <i>View</i> folders, which enable you to group <i>Search View</i> folders as best suit your requirements.</p> <ul style="list-style-type: none"> • A <i>Search View</i> is a folder of elements or structures that you assemble by assigning a model search to the folder; when you double-click on or expand the folder, the search runs and refreshes the folder contents • You can also set a search to refresh at a defined interval, and to notify you if new results are found 	<p>Model Views Operations ⁶⁰⁴</p> <p>Diagram Slide Show ⁶⁰⁷</p>

Topic	Detail	See also
	<p>Under a <i>Model View</i> root node you can create Views folders that can contain Search View folders, <i>Slideshow</i> folders and <i>Favorites</i> folders.</p> <ul style="list-style-type: none"> • A <i>Slideshow folder</i> contains diagrams only, which you can display as a slide show with diagrams being shown in the sequence in which they are listed in the folder. One folder represents one slide show - you can run the slide show automatically or manually; in either case the diagrams are closed after they have been displayed. • A <i>Favorites folder</i> gives you easy access to commonly-used items in the Project Browser; to create hyperlinks in a Favorites folder to the required items in the Project Browser, drag items from the Project Browser into the Favorites folder <p>You can also export all of the View folders containing Views from any root section as an XML file, and import a <i>Views XML file</i> as an additional, editable <i>Model View</i> root node.</p> <p>These are single-level items; if you drag a package into the Favorites folder, you cannot expand that package there.</p> <p>To select specific items inside a package, expand it in the Project Browser and then drag the items into the Favorites folder.</p>	
<p>Key to Model View Contents</p>	<p>Items on the model view can be identified by their accompanying icons:</p> <ul style="list-style-type: none">  - A Model View root node  - A View Folder  - A Favorites Folder  - A View (search-based)  - A Slideshow Folder  - The My Views root node  - The DoDAF-MODAF View (a Technology-defined View)  - The Recent Discussions folder  - A repository of postings from a selected team review, concerning elements in the Model Views  - The recent posts folder for the selected team review 	

Topic	Detail	See also
Accessing Operations	<p>To perform operations on the Model View contents, you can use the:</p> <ul style="list-style-type: none"> • Model Views window Toolbar or the • Item context menus <p>Each level of the Model Views hierarchy has a slightly different context menu.</p>	<p>Model Views Toolbar ⁶⁰⁰</p> <p>Model Views Context Menus ⁶⁰¹</p>

4.1.4.1 Model Views Toolbar







Access **View | Model Views > Toolbar**

Reference



The availability of the **Model Views** toolbar options depends on the type of object selected. The options are, from left to right:

Icon	Usage	Shortcut	See also
	Displays the appropriate Properties dialog for the item selected	Double-click on the item, or Enter	
	Locates the selected object in any diagrams in which it has been used in the model, and either: <ul style="list-style-type: none"> • Displays the single diagram with the object highlighted or • Lists the several diagrams in which the object has been located 		
	Locates and highlights the selected object in the Project Browser		
	Creates a new <i>Model View</i> root node, and displays the New Model View dialog in which you enter the root node name		
	Creates a new <i>Views</i> folder in the currently-selected root node		
	Creates a new <i>Favorites</i> folder in the currently-selected <i>Views</i> folder		

Icon	Usage	Shortcut	See also
	Creates a new Slideshow folder in the currently-selected <i>Views</i> folder		Diagram Slideshow ⁶⁰⁷
	Creates a new <i>View</i> in the currently-selected <i>Views</i> folder, and displays the Create New View dialog to define the search that populates the View		Model Views Operations ⁶⁰⁴
	Refreshes the selected <i>Model Views</i> root node, folder, <i>View</i> or <i>Favorites</i> ; for a <i>View</i> , this runs the Model Search defined in the View properties		Model Views Operations ⁶⁰⁴
	Moves the currently-selected object up or down within its type ; you cannot move - for example - a package below a diagram, or a <i>View</i> above a <i>Favorites</i> folder		
	Displays a prompt to confirm deletion of the selected object and - if appropriate - its contents You cannot delete the original <i>Model Views</i> , <i>My Views</i> or <i>Recent Discussions</i> root nodes, or any technology-defined <i>Views</i>		
	Displays Help on Model Views		

4.1.4.2 Model Views Context Menus

Access **View | Model Views: Right-click on object**

Reference

The **Model Views** window context menus display different options, depending on which level of the **Model Views** hierarchy you right-click on. The options are described below:

Menu Option	Action	Shortcut	See also
Properties	(Not for the <i>My Views</i> , <i>Recent Discussions</i> , initial <i>Model Views</i> or <i>Technology-defined</i> root nodes.) Display the appropriate Properties dialog for the selected object. You can edit any of the properties, if required; changes to objects populated from the model are reflected in all other views (Properties window, diagrams, reports) of that object. The Properties option for the <i>Recent Posts</i> folder		Diagram Slide Show ⁶⁰⁷

Menu Option	Action	Shortcut	See also
	<p>displays the Recent Post Options dialog, which enables you to specify the number of days back from which to extract postings from the Team Review, and the number of postings to list.</p> <p>The Properties option for a <i>Slideshow</i> folder displays the Slideshow Properties dialog, which enables you to automate the slide show and set the number of seconds for which each diagram is displayed.</p> <p>If you deselect the Enable checkbox, you must press (Spacebar) to display each diagram.</p>		
New Views Folder	<p>(Root node only.)</p> <p>Display a prompt for the <i>Views folder</i> name and create the folder in the selected root node.</p>		
Import Views From XML	<p>(Root node only.)</p> <p>Prompt for the XML file location and create a new <i>Model Views</i> node to hold the imported Views.</p>		Model Views Operations ^[604]
Export to XML (Views Only)	<p>(Root node only.)</p> <p>Prompt for a file path and name, and copy all Views under the selected root node to an XML file at that location.</p>		Model Views Operations ^[604]
Remove Model View	<p>(Not for the <i>My Views</i>, initial <i>Model Views</i> or <i>Technology-defined</i> root nodes.)</p> <p>Display a prompt to delete the selected user-defined Model View and, if confirmed, delete the root node and all contents.</p>		
New Search Folder	<p>(Viewfolder only.)</p> <p>Display the Create New View dialog (similar to the View Properties dialog) for you to define the search that populates the View.</p>		Model Views Operations ^[604]
New Favorites Folder	<p>(Viewfolder only.)</p> <p>Display the Create a new favorites based folder dialog, which prompts for the folder name.</p>		
New Slideshow	<p>(Model View, Viewfolder only.)</p> <p>Display the Create a new slideshow dialog, in which you type the name of the slide show.</p>		Diagram Slide Show ^[607]

Menu Option	Action	Shortcut	See also
	You must use the Properties dialog to define the properties of the slide show.		
Open Search	(View only.) Display the Model Search tab in the main work area, listing the full results of the search and giving access to all the facilities of the Model Search.	Shift+Space	Customizing the Search View ^[619]
Refresh	Refresh the search and open the View or Postings Repository to show the elements or Posts retrieved by the search.	Space	
Double Click Opens Search	Enable you to perform the Open Search function by double-clicking on the View. If you deselect this option, double-click refreshes the search and opens the View to show the elements retrieved by the search.		
Find Post	Open the Project Team Review and highlight the selected post.		
Open Team Review	Open the Project Team Review at the top level (Category).		
Edit Connections	Display the Team Review Server Connections dialog, to select which review to open.		Team Review Connections ^[296]
Remove Folder	(View folder only.) Display a prompt to delete the selected Views folder and, if confirmed, delete the folder and all contents.		
Remove View	Display a prompt to delete the selected View and, if confirmed, delete the View and all contents.		
Remove Favorites	Display a prompt to delete the selected <i>Favorites</i> folder and, if confirmed, delete the folder and all contents.		
Remove Slideshow	Display a prompt to confirm deletion of the selected slide show and, if confirmed, delete the slide show and all its diagrams.		

Menu Option	Action	Shortcut	See also
Run Slideshow	Run the slide show in the Diagram View.		Diagram Slide Show ^[607]
Run Slideshow Full Screen	Run the slide show in full screen mode, so that the slide show fills the whole screen.		Diagram Slide Show ^[607]
Stop Slideshow	Cancel execution of the slide show running in the Diagram View.	Esc	Diagram Slide Show ^[607]
In Project Browser	(An element, diagram or package object only.) Highlight the selected item in the Project Browser.		
In Diagrams	(An element or child Package object only.) Locate the selected object in any diagrams in which it has been used in the model, and either: <ul style="list-style-type: none"> • Display the single diagram with the object highlighted or • List the several diagrams in which the object has been located 		
Remove Linked Item	(An element, diagram or package object only.) Display a prompt to delete the selected object and, if confirmed, remove the object from the folder. This has no effect on the object in the Project Browser or any diagrams. You would not delete an object in a View, as it is replaced the next time the View is refreshed.		
Help	Display Help on Model Views.		

4.1.4.3 Model Views Operations

This topic describes the operations you can perform in the Model Views window.

Access [View | Model Views](#)

Topics

Topic	Detail	See also
<p>Define View Search</p>	<p>When you:</p> <ul style="list-style-type: none"> • Create a View, the Create New View dialog displays • Display the View properties, the View Properties dialog displays <p>These dialogs are identical.</p> <p>In the Name field, type a name for the View.</p> <p>In the Search field, either:</p> <ul style="list-style-type: none"> • Click on the drop-down arrow and select an existing search from the lists, or • Click on the (...) (Browse) button to display the Manage Searches dialog, edit an existing search or define a new one, then Close the dialog and select that search name in the Search field <p>A custom SQL search statement should return the <i>guid</i> and <i>type</i> of the object found so that Enterprise Architect can search for the selected item in the Project Browser</p> <pre>SELECT ea_guid AS CLASSGUID, Object_Type AS CLASSTYPE, Name FROM t_object</pre> <p>If required, in the Search Term field type a specific value to search for.</p> <p>If you are working with the Corporate or extended versions of Enterprise Architect, and you want the search to run automatically and refresh the results, select the Refresh this search checkbox.</p> <p>The Frequency field has three sections, for hours, minutes and seconds; click on the appropriate section and use the up and down arrows at the end of the field to set the interval for refreshing the search results.</p> <p>You can also set the refresh to display a pop-up notification if the search results change. To do this, select the Notify me when new results found checkbox.</p> <p>Click on the OK button; the View is created (or updated) in a collapsed state.</p> <p>When you expand the View, the search executes and populates the View.</p>	<p>Create & Modify Searches⁶²↑</p>
<p>Display Recent Postings</p>	<p>If a new post is added to the Team Review, or an existing post is updated, that post is also added automatically to the <i>Recent Discussions</i> <Connection> <i>Recent Posts</i> folder in the Model View.</p> <p>To open a post, double-click on the entry; the Team Review tab displays, showing the selected message.</p> <p>You can control how many posts are displayed, and for what period of time, using the Model Views context menu for the <i>Recent Discussions</i></p>	<p>Model Views⁵⁹↑</p> <p>Model Views Context Menu⁶⁰↑</p>

Topic	Detail	See also
	folder.	
Move Objects Into Favorites	Drag any required package, diagram or element from the Project Browser into the required <i>Favorites</i> folder.	
Move Objects Between Views	<p>Views and <i>Favorites</i> folders are fixed in the <i>Views</i> folder in which you create them, and you cannot move them.</p> <p>However, you can:</p> <ul style="list-style-type: none"> • Copy (by dragging) objects from any <i>View</i> into any <i>Favorites</i> folder, and • Move (by dragging) objects between any two <i>Favorites</i> folders 	
Use Objects From Model Views	<p>To make use of an element, diagram or package held in any <i>View</i> or <i>Favorites</i> folder, click on the item and drag it into a diagram or a Team Review posting</p> <p>The item behaves in the same way as if you dragged it from the Project Browser.</p>	Team Review 279
Export/Import Views	<p>You export Views to create an XML file that you can:</p> <ul style="list-style-type: none"> • Import into another model as a user-created Model View or • Call from an MDG Technology Selection (MTS) file to access the Technology-defined View provided by the active MDG Technology <p>The export and import functions are available from the Model Views root-node context menus.</p> <p>When you use the export function, it acts on the complete set of View folders in the selected <i>My Views</i> root node, <i>Model Views</i> root node, or user-generated root node; you cannot export individual Views, nor can you export Favorites folders.</p> <p>The function displays the Save As dialog, on which you browse for the directory location for the exported XML file, and specify the file name.</p> <p>When you use the import function, it displays the Select Import Filename dialog on which you browse for the directory and XML file you want to import; the import creates a new Model View folder with the same name as the copied root node.</p>	Model Views Context Menus 601
Set Up a Technology-Defined View	<p>To set up the Technology-defined View for an MDG Technology, you:</p> <ol style="list-style-type: none"> 1. Create a user-generated Model View in Enterprise Architect while using the technology. 2. Populate it with the required View folders and Views. 3. Export the Views from that Model View as an XML file to an appropriate location. 	Working with MTS Files 1335

Topic	Detail	See also
	<p>4. Create a call to the file from the technology's MTS file.</p> <p>Thereafter, any model for which the MDG Technology is active automatically displays those Views in a Technology-defined View.</p>	

4.1.4.4 Diagram Slide Show

The Slide show facility in the Model Views window enables you to create and run screen-based presentations of sets of diagrams:

When you run a slide show, it displays the diagrams in the folder in the sequence in which they are listed in the folder. After each diagram has been displayed, it is closed.

Access [View | Model Views](#)

Use to

- Create a slide show as a folder within a Views folder under a Model View node
- Run a slide show manually
- Run a slide show automatically with each diagram displaying for a period that you define
- Display a slide show within the Enterprise Architect work area
- Display a slide show in full screen mode
- Delete a slide show

How to

To create a slide show

Step	Action	See also
1	Under a Model Views node, click on the Views folder to contain the slide show.	
2	<p>Either:</p> <ul style="list-style-type: none"> • Right-click on the Views folder and select the New Slideshow option or • Click on the New Slideshow folder icon in the Model Views toolbar <p>The Create a New Slideshow Folder dialog displays.</p>	
3	In the Name field, type the name of the slide show.	
4	Click on the OK button.	









Step	Action	See also
	The new slide show folder is added to the selected View folder.	
5	Dock the Project Browser window separately from the Model Views window.	Dock Windows ^[135]
6	<p>Drag the diagrams to be displayed as part of the slide show from the Project Browser into the new slide show folder.</p> <p>Organize the diagrams in the order in which they are to be displayed, using the up/down green arrows in the Model Views toolbar.</p> <p>To remove a diagram that is not required, right-click on it in the slide show folder and select the Remove Linked Item context menu option.</p> <p>The diagram items in the slide show folder are links to the diagrams in the Project Browser, not copies of the diagrams or the diagrams themselves; any actions you take in the slide show have no impact on the original diagrams.</p>	Model Views Toolbar ^[600]
7	Close the folder.	

To define how the slide show is to operate, automated or manual, follow the steps below

Step	Action	See also
1	<p>Right-click on the slide show folder and select the Properties context menu option.</p> <p>The Slideshow Properties dialog displays.</p>	
2	<p>To automate the slide show, select the Enable checkbox.</p> <p>To run the slide show manually, deselect the checkbox.</p>	
3	If you are automating the slide show, in the Time field type the number of seconds that each diagram is to remain on display.	
4	Click on the OK button.	

To run a slide show of diagrams, either in the Diagram View or full screen

Step	Action	See also
1	Right-click on the slide show folder and select either the:	

Step	Action	See also
	<ul style="list-style-type: none"> • Run Slideshow option, to run the slideshow in the Diagram View • Run Slideshow Fullscreen option to run the slideshow using the full screen <p>The first diagram in the slide show displays.</p>	
2	<p>If you have set up the slide show to run automatically, you can leave it to display the diagrams as defined.</p> <p>You can also moderate the slide show using manual commands.</p>	
3	<p>If you are controlling the slide show manually, right-click on a slide.</p> <p>The following toolbar displays:</p> 	
4	<p>Control the slide show using the toolbar icons and other aids, as follows:</p> <ul style="list-style-type: none"> • Display the next slide - , (Spacebar) or (↑) • Display the previous slide - , (←) • Display the first slide - , (↑) • Display the final slide - , (↓) • Pause the slide show -  • Resume the slide show -  • Stop the slide show -  or, in Diagram View, right-click on the slide show folder and select the Stop Slideshow option 	

To delete a slide show

Step	Action	See also
1	<p>Right-click on the slide show folder and select the Remove Slideshow context menu option.</p> <p>Enterprise Architect prompts you to confirm the deletion.</p>	
2	<p>Click on the Yes button.</p>	

Step	Action	See also
	The folder and its list of links to diagrams in the model is removed.	

4.1.5 The Pan & Zoom Window

The Pan & Zoom window provides a 'birds-eye' view of diagrams. It enables you to navigate quickly around large diagrams.

Access [View | Pan & Zoom](#)

Reference

Topic	Detail	See also
View	The window shows a reduced image of the whole diagram. The shaded box represents the viewed area, displayed on the open diagram.	
Toolbar	The toolbar provides the following functions (in order): <ul style="list-style-type: none"> • Zoom In • Zoom Out • Zoom to fit diagram • Zoom to fit page • Zoom to 100% • Zoom Slider 	
Use	Open the required diagram. Move the cursor onto the shaded box inside the Pan & Zoom window. Hold down the mouse button as you move the cursor to move the shaded box over the open diagram. To zoom, use either the Zoom Slider or the buttons located on the tool bar.	

4.2 Search: Finding Information



Searching involves using ad hoc search terms and various search types to locate elements and information within your model, regardless of location or context.

The result sets can be used to manage elements, to create documentation or to find the location of elements within the complete model structure.

Topics

Topic	Link
Model Search - how to use the searching capability within Enterprise Architect to find and display various kinds of information	Model Search ^[611] Model Search Context Menu ^[614] Model Search Toolbar ^[616] Pre-defined Searches ^[617] Customizing the View ^[619]
Create and Modify Searches - how to modify and work with search definitions	Create & Modify Searches ^[621]
Adding Filters - further modifications to searches	Add Filters ^[627]
Diagram Filters- how to suppress or highlight information in a diagram using customizable filtering mechanisms based on element properties such as status and complexity	Diagram Filters ^[629]

4.2.1 Model Search

The Model Search allows you to quickly navigate and find elements within your model. It lists each element that meets the search criteria you specify within the search terms and search type. Many different kinds of searches are already built into Enterprise Architect, and it is possible to create very detailed and specific searches of your own. It is even possible to export and share your favorite searches.

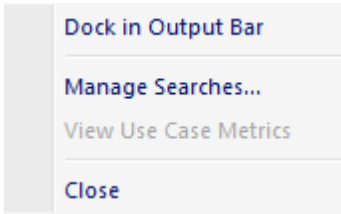
Access [Edit | Model Search](#) (**Ctrl + Alt + A**) or (**Ctrl + F**)

Use to

- Search for elements within your model based on various criteria
- Generate targeted reports based on your result set

- Manage result sets - for example set properties and delete elements

Field	Usage	See also
Search Term	Type in the word, phrase or characters to search on. Search filters enable you to perform customized searches on a Search Term in order to locate model elements having specific characteristics.	
Search	The Model Search window provides a set of <i>built in</i> searches for your convenience; the Search drop-down list provides several pre-defined searches including. <ul style="list-style-type: none"> • Simple • Extended • Find Orphans • Recently Modified Elements <p>In the drop-down list, the built in searches are followed by searches that:</p> <ul style="list-style-type: none"> • You have created yourself • Your organization has created for use within the project • Have been created as part of workflow script • Have been created in the Enterprise Architect Scripting facility 	Pre-defined searches ^[617] Functions - Create a Search With User Tasks ^[307] Workflow Data Structures You Fill ^[303] Script Group Properties ^[2375]
Run	Run the search using the entered text and selected Search.	
Options	The Options button displays the Search Options submenu, which enables you to display the search results as a tab of the Output window rather than in the Model Search View. An advantage of moving the search results to the Output window is that you can select items from the search results and drag them onto a diagram, which you cannot do when the results are in the Model Search View. If you select the Dock in Output Bar menu option, when you next display the menu this option becomes Dock in Main View . The Search Options submenu also provides the means of performing advanced searches on your project, and displaying project metrics.	The System Output Window ^[164] Advanced Search Options ^[628] Estimating Project Size ^[520]

	 <p>The Options button also enables you to perform advanced searches.</p>	
Builder	Show or hide the Search Builder facility.	Create Search Definitions ^[623]
Result List	Display the results of your search.	

Keyboard Options:

- Display the properties of an element (**Double Click or Enter**)
- Drag an item(s) onto a diagram (**Ctrl + drag a selected item**)
- Toggle between the Notes and the Model Search window (**Ctrl + Shift + Enter**)
- Open the Linked Document window for the selected Element (**Ctrl + Alt + D**)
- Select All Items (**Ctrl + Alt + A**)

Notes

- You can access the Model Search facilities and perform specific searches indirectly, from Add-Ins, from MDG Technologies, from a hyperlink and from a shortcut to access your model; this entails setting up a search profile either in the appropriate tool, or as an XML file accessed by the tool

Searching from	See
An MDG Technology Selection (MT) File (using an exported search definition)	Working With MTS Files ^[1335] Create & Modify Searches ^[621]
A Login Shortcut	Project Shortcuts ^[183]
An Add-In	Add-In Search ^[2594]
A Hyperlink	Hyperlinks ^[1668]

- Running a custom or diagram based search disables some context menu options

4.2.1.1 Model Search Context Menu

You can select elements or diagrams in the Model Search and perform various operations on them, as well as simply dragging the item into a Team Review post.

Access Results pane, right-click on an element

Use to

- Generate reports from search results
- Print search results
- Copy and work with search results
- Manage elements returned by a search

Reference

Menu Option	Action	Shortcut	See also
Scripts	<p>Displays a list of any Model Search scripts you have created, which can be executed on the selected item.</p> <p>This option displays only if:</p> <ul style="list-style-type: none"> • You have created Model Search scripts in a Model Search Script Group folder, and • The item results from a successful Query search, or • The item results from a successful SQL search that includes CLASSGUID and CLASSTYPE 		Script Group Properties ^[2375]
Properties	Display the Properties dialog for the element.		
Copy Selected to Clipboard	Copy the selected item to the MS Windows clipboard so that it can be pasted to a document, spreadsheet or email.		
Documentation	<p>Generate an RTF report; you have two options:</p> <ul style="list-style-type: none"> • Generate a separate report on each selected object in the Model Search 		Generate RTF Documentation Dialog ^[2256]

	<ul style="list-style-type: none"> Generate one report on all selected objects <p>In either case, the Generate RTF Documentation dialog displays.</p> <p>If you generate the report using a custom SQL search, the SQL must include <i>ea_guid</i> AS <i>CLASSGUID</i> and the object type.</p>		Create Search Definitions [623]
Create Linked Document (Edit Linked Document)	Create (or edit) a linked document (Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions).	Ctrl + Alt + D	Link Document to UML Element [927] Edit Linked Documents [930] Linked Documents [928]
Delete Linked Document	Delete an existing linked document. (This option only displays if the element has a linked document.)		
Print	Print out the filtered results.		
Clear Results	Clear the search results from the Model Search.		
Find in Diagrams	Display the diagram that uses the element or, if the element is used in multiple diagrams, display a list of diagrams to choose from.		
Find in Project Browser	Highlight the element in the Project Browser.		
Bookmark Selected	Bookmark the element.		
Delete Selected	Delete the selected element from the Model Search.		
Close	Close the Model Search.		
Help	Display this Help topic on the Model Search.		

--	--	--	--

Learn more

- [Add a New Post](#)^[287]

4.2.1.2 Model Search Toolbar

The Model Search toolbar displays at the top of the Model Search window, enabling you to operate on the results of your model search.

Access **Edit | Model Search**

Use to

- Edit the text of notes for an item
- Delete selected items from the model
- Print the search results
- Generate and print an RTF report on selected items
- Hide or abbreviate notes text in the output
- Hide or show the filter bar under the column headings
- Display Help text

Reference



Action	Usage	Shortcut	See also
Edit Notes	For the selected item, open the Notes window (if it is not already open) so that you can edit the text of the notes.	Ctrl + Spacebar	Notes ^[977]
Delete	For a selected item or group of items, delete them from the model; refresh the project to check that the items have been deleted.	Ctrl + D	
Print	Print the complete set of search results		
Rich Text Report	For a selected item or group of items, generate and print an RTF report.		Generate RTF Documentation Dialog ^[2256]
View Notes	Display a short menu that enables you to select		

	<p>whether, for all items, to:</p> <ul style="list-style-type: none"> • Hide any Notes text from display in the search results • Display the first few words of the Notes text in the search results • Display the full Notes text in the search results 		
Toggle Filter Bar	Hide or reveal the filter bar underneath the column headings.		Customizing the Search View ^[619]
Help	Display the Enterprise Architect Help, starting with the Model Search Help topic.		

4.2.1.3 Pre-defined Searches

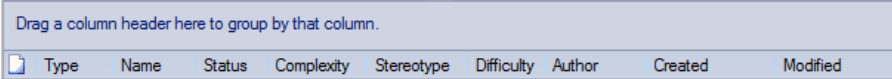
The following pre-defined searches are provided with Enterprise Architect, and are listed in the *Built-In* category in the **Search** drop-down list.

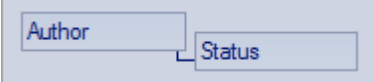
Search	Description	See also
Simple	Searches the Name , Alias and Notes fields of all elements for the given search term.	
Extended	Searches many additional fields relating to the element, including Attributes , Operations , Tagged Values and Test Cases .	
Element Name	Searches for an exact match against the element name, alias, method or operation; the default search in the Search in Model menu option in the Code Editor.	Code Editor Context Menu ^[1805]
Attribute Details	Searches for elements with attributes relating to the search term, including Tagged Values, constraints, and common attribute data fields.	
Find Orphans	Searches for orphaned elements throughout the model, with the ability to filter on common element fields using a search term. An 'orphaned' element is an element that does not appear on any diagram in the model.	
Failed Internal Tests	Searches for elements containing internal test cases where the search term is in any common Test Case field	

	and the Status value is <i>Fail</i> .	
Method Details	Searches for elements with operations and methods relating to the search term, including Tagged Values, constraints and common operation and method data fields.	
Responsibility	Searches for elements with internal responsibilities/ requirements where the search term relates to any common Responsibility/Requirement field.	
Resources	Searches for elements with assigned resources where the search term relates to any common Resource field.	
Requirements	Searches for Requirement element types where the search term relates to any common element field.	
Find Bookmarked Elements	Searches for elements that have been bookmarked, anywhere in the project.	
Recently Modified Elements	Searches for elements that have been recently modified, anywhere in the project. The search term relates to any common element field. The default is to show elements modified in the last three days, but you can set an alternative interval by typing the appropriate number of days in the Search Term field.	
Recently Modified Diagrams	Searches for diagrams that have been recently modified, anywhere in the project. The search term relates to any common diagram properties field. The default is to show diagrams modified in the last three days, but you can set an alternative interval by typing the appropriate number of days in the Search Term field.	
My Checked Out Packages	Searches for packages that are marked as checked out by the currently-logged in user.	

4.2.1.4 Customizing the Search View

This topic explains how to customize the display of search results after they are generated.

Topic	Detail	See also
Sort and Select	<p>In the Model Search you can:</p> <ul style="list-style-type: none"> • Sort the items by any column value in ascending or descending order, by clicking on the column header • Display element or diagram properties, by double-clicking on the item • Select: <ul style="list-style-type: none"> • An element or diagram by clicking on it • Several individual elements or diagrams by holding (Ctrl) as you click on them • A range of elements or diagrams by holding (Shift) as you click on the first and last in the range • All elements or diagrams in the list by pressing (Ctrl+A) 	
The View Header	 <p>The View Header defines the columns of information that are presented by the Model Search, and the order in which data items are presented.</p> <p>To review the column headings:</p> <ul style="list-style-type: none"> • Right-click on the header to display the Field Chooser context menu option; the Field Chooser dialog displays <p>This enables you to add columns to or remove columns from the output; between them, the View Header and Field Chooser dialog show the full range of column headers available.</p> <p>To add a column heading to the View Header:</p> <ul style="list-style-type: none"> • Drag it from the Field Chooser dialog onto the header to the position you want the column of data to display • When you have selected the column headings you require, click on the red cross in the top right corner of the Field Chooser dialog to close it <p>To remove a column from the output:</p> <ul style="list-style-type: none"> • Drag the column heading downwards out of the View Header <p>To change the sequence of columns:</p>	

Topic	Detail	See also
	<ul style="list-style-type: none"> • Drag the required column header left or right, as appropriate 	
<p>Value Grouping</p>	<p>You can organize the reported data hierarchically according to the value of one or more of the column categories.</p> <p>You might organize the data by Type, and within each Type by Name; if you then click on any of the other column headings, the data within this grouping is further sorted with the values of the selected column (for example, Created) in ascending or descending order.</p> <p>To set up the value grouping, drag the column heading representing the primary grouping (such as Author) onto the Drag a column header here to group by that column field.</p> <p>Now drag the column heading for the next level of grouping (such as Status) to the right of the first heading; the two heading titles display as connected blocks, as shown below:</p>  <p>You can, if required, add further levels of grouping by dragging other column headings onto the hierarchy (such as Created), and restructure the order by dragging existing or additional headings into the level you want them to hold.</p> <p>For example, you could make Type the secondary grouping by dragging it to the right of Author, or drop Status between Author and Type.</p> <p>To remove a grouping level, drag the appropriate column heading out of the sequence and below the View header; any subordinate groupings move up a level.</p>	
<p>Filter Bar</p>	<p>The Filter Bar displays underneath the column headings; you can hide or show it using the Toggle Filter Bar icon in the toolbar.</p> <p>The Filter Bar consists of a row of fields, one field per column; as you type a value into one of these fields, the Model Search results are automatically filtered to show entries with a value in the corresponding column that match the value in the field. The filter is case sensitive.</p> <p>For example, if you type the letter F in the Filter Bar field under the Author column, the results immediately reduce to entries for which the author name includes names such as Fred, Frank, Felix, Felicity, Fairmile, Farmer or Fitzgibbon (but none where the name includes f only as a lower case letter, such as Rufus).</p> <p>If you then go on to type Fi, the results immediately reduce again to only entries for which the author name includes names such as Fitzgibbon.</p>	

4.2.1.5 Create & Modify Searches

You provide search filters and create new search definitions using the Search Builder tab, displayed underneath the **Search Term** and **Search** fields on the Model Search window.

Access [Edit](#) | **Model Search: Builder**

Use to

- Create new search definitions and edit them

Search Builder Toolbar:

The search builder toolbar enables you to configure the system-provided searches, and to create and edit your own searches

The toolbar icons, from left to right, provide access to the following functions:

Option	Usage	See also
New Search	Create a new search definition, with new search criteria.	Create Search Definitions ^[623]
Save Search	Save a modified or new search.	
Copy Search	Copy the existing search selected in the Search field, to modify. You can copy: <ul style="list-style-type: none"> • Searches you or other users have created • Predefined searches and edit the copies, but you cannot edit the predefined searches themselves 	
Restore Default	Restore any changed parameters to the default settings and format.	
Delete Search	Delete the search definition from the Search drop-down list.	
Add Filter	Add a new set of parameters to filter the search on.	Add Filters ^[627]
Remove Filter	Delete the selected filter from the search.	
Export Search	Display a selection box that enables you to select searches to export to an external directory as an XML Search file.	
Import Search	Display the Windows Directory Explorer Open dialog to enable you to import searches as XML Search files from an external directory.	

Option	Usage	See also
Search Options	Display the Advanced Options dialog, to define where a query-builder search should operate and how the search should match filters.	Advanced Search Options ^[626] Create Search Definitions ^[623]

Search Builder:

You use the main body of the Search Builder tab to configure the element search filters that are contained in the selected search.

The Search Builder tab defaults to a **Simple** search, which changes as you select to create a new search or select the search to modify.

Column	Usage	See also
Search In	Select the type and name of each element feature (the element field name) to search on.	
Condition	Select the condition of the search parameter (the condition placed on the field value). The available options are Contains, Equal To, Not Equals and One Of .	Fields and Conditions ^[629]
Look for	Specify the search term (the actual value or delimiting value) to perform the conditional search on. This value can pertain to the selected element type; for example, the value could be a date for <i>DateCreated</i> or a text value for other fields. The search term can contain multiple values, separated by commas.	
Required	Indicate whether the field is mandatory - that is, whether the search results must include elements with your search term in that field. The following provides example of using the Required checkbox, illustrated by the default Simple search, which checks the Name, Alias and Notes fields of all elements: <ul style="list-style-type: none"> • If the Required checkbox is not ticked for any field, then if the search term is found in the Name, Alias OR Notes field for an element, that element is listed in the results • If the Required checkbox is ticked for all fields in the search, then the search term must be found in all of those fields before the element is listed in the results; that is: Name AND Alias AND Notes • If the Required checkbox is ticked for some fields but not others, the search term must be found in all fields for which the checkbox is 	Advanced Search Options ^[626]

Column	Usage	See also
	<p>ticked, and <i>at least one</i> of the fields in which the checkbox is not ticked; that is: Name AND (Alias OR Notes)</p> <p>This checkbox helps you to define the filters on a specific feature (such as Element, Diagram or Attribute). You can also set up a filter to check for a defined value in every feature covered by a search (AND filter) or in any feature covered by the search (OR filter).</p>	

4.2.1.5.1 Create Search Definitions

User-defined searches are created using the Search Builder tab, and are stored in the Program Files directory (not in the project repository).


Access [Edit | Model Search: Builder](#)

Use to

- Create a search definition using the SQL Editor, Query Builder, or an Add-In Search

How to

To create a new search definition

Step	Action	See also
1	Click on the New Search icon in the toolbar. The Create New Search Query dialog displays.	
2	In the Search Name field, type a name for the search.	
3	<p>Select the radio button for the type of search you require:</p> <ul style="list-style-type: none"> • Query Builder - provides an interface that enables you to design your own search • SQL Editor - enables advanced users to directly write SQL Select statements • Add-In Search - enables you to supply the name of your Add-In and a method (for example <i>MyAddin.RunThisMethod</i>), which is called whenever the search is run; this search can be exported and distributed as a part of your Add-In 	Add-In Search 

Step	Action	See also
4	Click on the OK button, and refer to the following table.	

Search Types:

Search	Description	See also
Query Builder	<p>Your search definition appears as being selected in the Search drop-down field.</p> <p>You can now click on the Add Filter toolbar icon to add filters.</p>	Add Filters [627]
SQL Editor	<p>The Custom SQL dialog displays, enabling you to input your <i>SELECT</i> statement.</p> <p>The SQL editor is based on the common Code Editor, and provides an intellisense autocompletion list populated with Enterprise Architect's repository structure; to display the autocompletion list, press (Ctrl+Spacebar).</p> <p>A simple search might be to locate an object from an Enterprise Architect table, given a search term that the user enters in the Model Search Search Term field</p> <ul style="list-style-type: none"> <i>SELECT * FROM t_object WHERE NAME='<Search Term>'</i> <p>Enterprise Architect also enables you to use #xxx# macros as string replacers in <i>WHERE</i> statements, so that the same search can be used by different people in different environments; these macros include:</p> <ul style="list-style-type: none"> #WC# - Gets the appropriate wild card for the current database, and so enables the search to be performed on models on different databases <p><i>t_object.Name LIKE '#WC#Test#WC#'</i></p> <ul style="list-style-type: none"> #Author# - Takes the value of the Author field in the Options dialog General page, and enables the defined search to be performed on objects created by that user (this value can be manually re-set in the Options dialog) #DB=<DBNAME># where <DBNAME> can be one of the following: <ul style="list-style-type: none"> MYSQL JET ORACLE SQLSVR ASA OPENEDGE 	Code Editor Functions [1809]

- POSTGRES

For example:

```
#DB=ORACLE# t_object.ModifiedDate >= (SYSDATE - INTERVAL
'<Search Term>' DAY)
```

- **#UserName#** - Gets the name of the person logged into version control

```
t_package.PackageFlags LIKE
'#WC#VCCFG=#WC#CheckedOutTo=#UserName##WC#'
```

(this is from Enterprise Architect's built in search *My Checked Out Packages*)

- **#Package#** - gets the currently-selected package's package_ID
- **#Branch#** - gets the IDs of the child packages of the currently-selected package, working recursively down to the lowest level of sub-package

For all Enterprise Architect functions in which you use a custom SQL statement (such as RTF reporting or Model Views) the statement must return the *guid* and *type* of the object found so that Enterprise Architect can search for the selected item in the Project Browser


```
SELECT ea_guid AS CLASSGUID, Object_Type AS CLASSTYPE, Name
FROM t_object
```

You can extend the usability of your SQL searches using the aliases *CLASSGUID* and *CLASSTYPE*, which enable Enterprise Architect to display the Properties dialog and icon for elements, connectors, attributes or operations, as well as selecting them in the Project Browser.

Some simple examples for using these aliased fields are provided below:

- ```
SELECT ea_guid AS CLASSGUID, Object_Type AS CLASSTYPE,
Name FROM t_object
```
- ```
SELECT ea_guid AS CLASSGUID, Connector_Type AS CLASSTYPE,
Name FROM t_connector
```
- ```
SELECT ea_guid AS CLASSGUID, 'Operation' AS CLASSTYPE, Name
FROM t_operation
```
- ```
SELECT ea_guid AS CLASSGUID, 'Attribute' AS CLASSTYPE, Name
FROM t_attribute
```

When you have defined the *SELECT* statement, click on the **Save** button to save this search; the search is then available from the Search drop-down list.

Add-In Search	<p>Type in:</p> <ul style="list-style-type: none"> • The name of your Add-In • A period (full stop) and • The name of the method to be called (for example, <i>MyAddin.RunThisMethod</i>) <p>Your search is automatically saved and available from the Search drop-down list.</p>	Add-In Search 
----------------------	---	---

Learn more

- [Customizing the Search View](#) 
- [Create & Modify Searches](#) 

4.2.1.5.2 Advanced Search Options

The advanced search filter options help you to apply greater control over how your customized search operates on your model.

You can set values and checkboxes in your search definition to find specific criteria within a **feature**, for example, element properties. However, you can **also** set filters to run the search across the whole model or a specific package, and to locate objects that have a specified value in:

- **All** features (for example, the value exists in an element's properties **and** one of its attributes' properties) or
- **Any one** of its features (for example, the value exists in an element's properties **or** any of its attributes' properties)

The filters for each feature operate on a database table. In the default Enterprise Architect searches, for example, filters on element properties operate on the *t_element* table, filters on attribute properties operate on *t_attribute*, and filters on method parameters operate on *t_operationparams*.

Access [Edit | Model Search: Builder: Search Options \(toolbar icon\)](#)

Use to

- Define whether your search should operate on the whole model, or on a specific package within the model
- Define whether your search should return items that match the search term in **every** feature or **any** feature defined for that search

Reference

Action	Usage	See also
Return matching items for the entire model	<p>Run the search across the entire model (this is the default option).</p> <p>The search checks every package in the model, and returns items that match the search criteria from wherever they are located.</p>	
Return matching items for the selected Package	<p>Run the search on a specific package that you select from the Project Browser.</p> <p>If you run a search and then select a different package in the Project Browser, the search results do not change until you click on the Run button again; that is, to search different areas of the project:</p> <ul style="list-style-type: none"> • Click on the first required package in the Project Browser • Click on the Run button • Check the result • Click on another package in the Project Browser • Click on the Run button 	
All Tables	<p>List objects that have a match in every feature table checked in the search.</p> <p>For example, only list elements where both the element and one or more of its attributes have the same stereotype name.</p>	
At Least One Table	<p>List objects that have a match in at least one of the feature tables checked in the search (the default).</p> <p>For example, list elements that either have the specified stereotype name, or have attributes that have the specified stereotype name.</p>	

Learn more

- [Create & Modify Searches](#) ⁶²⁷

4.2.1.5.3 Add Filters

Access Project | Documentation | Rich Text Format (RTF) Report: Add Filter
 Edit | Model Search: Builder: Add Filters (toolbar icon)

Use to

- Add filter criteria to a model search or an RTF report template

Reference

Option	Action	See also
<p>Search On</p>	<p>Select items to build up search filters on any information about an object.</p> <p>The following is a list of what is available, before you have defined a search:</p> <div data-bbox="438 488 868 1084" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <ul style="list-style-type: none"> Element Diagram Attribute Attribute.AttConstraint Attribute.AttTagValue Change Custom Property Constraint Method Method.MethodTagValue Method.Parameter Method.PostCondition Method.PreCondition Method.Parameter.ParamTagValue File Issue Defect Scenario TagValue Task Test Responsibility <li style="background-color: #0070C0; color: white;">Resource </div> <p>If you are adding filters to an existing search, the list contains only items appropriate to the initial filter.</p> <p>For example, if the initial filter is on diagram properties, the list for any subsequent filters for the search only contains the Diagram option.</p>	
<p>Include</p>	<p>Select each field item to include in your search (select the checkbox).</p>	
<p>Field</p>	<p>Identify the name of the field to search.</p> <p>The list presents items specific to the filter <i>Search On</i> item.</p>	<p>Fields and Conditions ^[629]</p>
<p>Condition</p>	<p>Specify the condition of the search parameter.</p>	
<p>Value</p>	<p>Type a value pertaining to the selected element field.</p> <p>For example, the value could be a date for <i>DateCreated</i> or a text value for other fields.</p> <p>The search term can contain multiple values separated by commas.</p>	
<p>Required</p>	<p>Select a particular field to generate a result set that <i>must</i> contain your search term in that field.</p>	

Option	Action	See also
Check All	Select all the items to include them in the search definition.	
Uncheck All	Deselect all the items to omit them from the search definition.	
OK	Apply the filter. The fields selected are added to the search definition.	

Notes

- You can add multiple search definitions as necessary; however, if you select the **Required** field in multiple definitions the search rapidly becomes impractical
- Multiple search definitions are better for 'and/or' searches

Learn more

- [Create Search Definitions](#) ^[623]
- [Create & Modify Searches](#) ^[621]
- [Generate RTF Documentation Dialog](#) ^[2256]

4.2.1.5.3.1 Fields and Conditions

When you click on a condition for a particular field, a selection of condition options becomes available. The list of options varies dependent on the type of information required.

For some conditions, the **Value** field contains an ellipsis (...). Click on this to display a selection dialog. This dialog varies dependant on the information required.

Learn more

- [Create Search Definitions](#) ^[623]
- [Add Filters](#) ^[627]

4.2.2 Diagram Filters

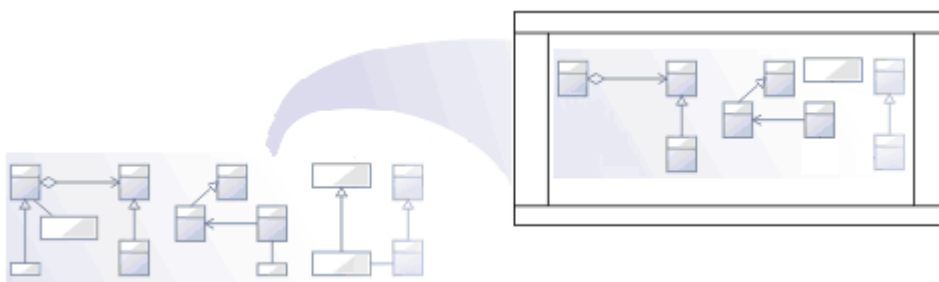


Diagram Filters (Dynamic Visual filters) enable you to modify the display of diagram components, so that the relevant items are immediately identified for the reader's attention without damaging the structure and

integrity of the model. Currently the feature operates on elements and connectors, and filters according to properties such as Author, Status, Date Created or Stereotype and, for connectors, whether the connector is hidden or visible.

Access [View | Diagram Filters](#)

Topics

Topic	Detail	See also
<p>Use</p>	<p>Use Diagram Filters to tailor the display of diagrams:</p> <ul style="list-style-type: none"> • For different users, so that - for example - technical staff and stakeholders each have a view that highlights the information pertinent to them • To show what elements have been recently developed or changed • To show which part of a model was developed by a particular person • To show which parts of a diagram are at a particular phase, status or version <p>Filters for elements can also be applied to the Package Browser and Diagram List; the <i>Select</i> filter effect does not, however, make a difference in these contexts.</p>	<p>Package Browser⁵⁸⁷</p> <p>Diagram List⁵⁹⁶</p> <p>Work With Diagram Filters⁶³⁷</p>
<p>Develop Diagram Filters</p>	<p>You create and define as many filters as you require, setting up each filter by defining which element or connector properties to specifically check for and (depending on how you set up the filter parameters) whether to include or exclude elements or connectors having particular property values.</p> <p>You cannot combine element and connector properties in one filter, but you can apply an element filter and a connector filter to a diagram at the same time.</p> <p>You can select to:</p> <ul style="list-style-type: none"> • Mute the irrelevant items in grayscale or a faded color • Hide the irrelevant items completely, or • Highlight (with a hashed line) the items that are relevant 	<p>Work With Diagram Filters⁶³⁷</p>
<p>Filter Action - Mute or Hide</p>	<p>If you select to Mute or Hide items, the action of the filters is to <i>exclude</i> items that <i>do not</i> match the parameters rather than <i>include</i> items that do.</p> <p>If, for example, you selected to filter on element name, looking for elements with the word <i>Class</i> in the name, the filter would apply the following logic:</p> <p style="padding-left: 40px;">Does Name contain string <i>Class</i>? If No, apply effect; if Yes, take <i>no</i> action.</p> <p>The elements you want are therefore what is left on the diagram, rather than what was operated on.</p>	<p>Work With Diagram Filters⁶³⁷</p>

Topic	Detail	See also
	The filter effect remains in force while you do other work on the diagram, until such time as you disable the filter.	
Filter Action - Highlight	<p>If you select the <i>Highlight</i> effect, the logic is reversed:</p> <p style="padding-left: 40px;">Does Name contain string <i>Class</i>? If No, take <i>no</i> action. If Yes, apply effect.</p> <p>In this case, the filter effect is not permanent, and clicking off the items deselects them.</p> <p>This effect is excellent for selecting items having the same characteristics across a large diagram, to be processed in a single task; instead of having to locate the items and select them with (Shift) +click individually, you can apply the filter.</p> <p>If you inadvertently lose the selection by clicking off the items, you can get it back again almost immediately by re-applying the filter.</p>	Work With Diagram Filters 631
Application	<p>You can use filters singly, in sequence, or in combination; for example, you could:</p> <ul style="list-style-type: none"> • Set up a filter for immediate use on a diagram, and modify that filter as you review the diagram so that you highlight elements with different values for the same properties - perhaps, by filtering on Phase, to compare 'as-is' and 'to-be' elements • Set up a filter and leave it active so that all diagrams you display are automatically filtered the same way • Set up a series of filters to use: <ul style="list-style-type: none"> • in one or more sequences to progressively highlight a diminishing set of items, or • alternately to highlight contrasting views of the diagram 	

Learn more

- [A Quick-Start Guide to Using Diagram Filters](#)

4.2.2.1 Work With Diagram Filters

The Diagram Filters window is a dockable window, which you can move around or fix next to the Diagram View while you activate, deactivate and edit the filters.

Access [View | Diagram Filters](#)

Use to

- Create and edit filters

- Set the effect of the filter
- Enable and disable filters
- Delete filters

How to

To create a filter to be applied to your diagrams

Step	Action	See also
1	<p>Perform one of the following actions:</p> <ul style="list-style-type: none"> • On the Diagram Filter toolbar, click on the New Filter icon - the first on the left • Right-click on the list panel and select the New Filter context menu option <p>The Create New Diagram Filter dialog displays.</p>	
2	<p>In the Enter Filter Name field, type a name for the filter, then click on the OK button.</p> <p>The Diagram Filter dialog displays.</p>	
3	<p>In the Search On field, either:</p> <ul style="list-style-type: none"> • Use the default value of Element or • Click on the drop-down arrow and select Connector <p>If you select Connector, the dialog changes to offer a smaller set of properties. The Visible option enables you to filter out hidden or visible connectors that meet other filter conditions that you set.</p>	
4	<p>Scroll through the properties to filter on, and select the checkbox against each property you require.</p>	
5	<p>For each property, click on the Condition field and select, from the drop-down list, the comparison condition to be applied.</p> <p>Consider how the combination of Condition - Equal To / Not Equal To and Filter Effect might affect the results on the diagram.</p>	Diagram Filters ⁶²⁹
6	<p>For each property, double-click on the Value field and type or select any specific value to filter on.</p>	
7	<p>Click on the OK button to save the filter and return to the Diagram Filters window.</p>	

Other Operations:

Topic	Detail	See also
Edit a Filter	<p>To edit an existing filter on the Diagram Filters window, either:</p> <ul style="list-style-type: none"> • Double-click on the filter name • Click on the filter name and select the Properties icon from the toolbar (the second icon from the left), or • Right-click on the name and select the Properties context menu option <p>The Diagram Filter dialog displays; adjust the filtered fields as described above.</p>	
Change the Name of the Filter	<p>Right-click on the name and select the Change Name context menu option.</p> <p>The Create New Diagram Filter dialog displays.</p> <p>Type over the existing name with the new name, and click on the OK button.</p>	
Set Effect of Filters	<p>To set how your filters identify selected items on your diagrams, click on the drop-down arrow of the toolbar Filter Effect field, and select one of the following options:</p> <ul style="list-style-type: none"> • Fade - display all items that do <i>not</i> match the filter criteria in a pale version of the diagram background color • Gray Scale - display all items that do <i>not</i> match the filter criteria in pale gray • Hide - conceal all items that do <i>not</i> match the filter criteria • Select - select and highlight (with a hashed line) all those items that <i>do</i> match the filter criteria 	
Enable Filters	<p>To enable a filter so that it takes immediate effect on your diagrams, select the check box against the filter name.</p> <p>You can select more than one filter at a time, to combine their effects.</p>	
Disable/ Clear Filters	<p>To disable a filter and clear the effect of the filter on the diagram so that it displays in full, clear the checkbox.</p> <p>To disable all filters, either:</p> <ul style="list-style-type: none"> • Click on the Reload Filters icon in the toolbar (third option from the left), or • Right-click on the list panel and select the Reload Filters context menu option 	

Topic	Detail	See also
Delete a Filter	To remove a filter from the Diagram Filters window, either: <ul style="list-style-type: none">• Click on the filter name and click on the Delete icon in the toolbar, or• Right-click on the filter name and select the Delete Filter context menu option	

4.3 Trace: Tracking Dependencies



Traceability helps to identify the way a given requirement or process has been implemented in a system, enabling you to follow the path of dependencies from the initial request, through a modeled solution and up to the deployed physical system or process.

Topics

Topic	Link
<p>Application - A well developed model provides full traceability in any direction - providing answers to questions such as "why was that particular hardware used in the solution"; assuming there is traceability back to a requirement that specified particular response times or functional capability.</p> <p>The result sets can be used to manage elements, to create documentation or to find the location of elements within the complete model structure.</p>	
<p>Tools Overview - There are various tools in Enterprise Architect that enable you to trace the definition and implementation of a process from initial requirement to generated code or technical deployment, or vice versa.</p> <p>A good starting point is to look at the topics listed to the right, which detail some of the more useful tools.</p> <p>The Traceability window, in particular, is designed to provide very detailed information about an elements immediate and distant relationships and dependencies.</p>	<p>The Traceability Window ^[637]</p> <p>Relationship Matrix ^[639]</p> <p>Gap Analysis Matrix ^[651]</p> <p>The Relationships Window ^[648]</p> <p>The Project Browser ^[567]</p> <p>Example Diagram ^[649]</p> <p>Traceability Tools ^[636]</p>
<p>Tracing Transformations - If you have performed any Transformations in developing your model and code, Enterprise Architect automatically creates <i>Transformation Dependency</i> connectors that you can trace - with the Traceability window - to establish what objects and code have been generated from each PSM element, or what the initial PSM element was for a generated object.</p> <p>Whether you use transformations or develop the stages of the model in other ways, you can build up a range of <i>Traceability diagrams</i> (<i>Custom</i> diagrams) to identify the development pathway and the dependencies between entities such as Requirements, Use Cases, Classes, Packages, Test Cases and other model artefacts, or possibly between these entities and the overall business process model.</p>	<p>Model Transformation ^[1679]</p> <p>The Traceability Window ^[637]</p> <p>Example (Traceability) Diagram ^[649]</p> <p>A Complete Business Process ^[1522]</p>

4.3.1 Traceability Tools

The model structure and a Traceability diagram act as starting points for tracing the definition, design and implementation of a feature of a system or process.

By applying tools such as the Relationship Matrix and Traceability window to these, you can follow threads throughout the model to determine how the feature is implemented and tested.

You can also obtain information on what elements realize and are realized by the elements in a given package, using the Dependency report and Implementation report, respectively.

Topic	Detail	See also
<p>Traceability Window</p>	<p>The Traceability window is a most useful and versatile traceability tool; starting with a Traceability diagram or a package structure in the Project Browser, you use the Traceability window to quickly explore the relationship chain of which any element is a component.</p> <p>When you click on an element, it immediately becomes the top point in the Traceability window.</p> <p>When you click on the background of a diagram, <i>all</i> elements in the diagram are listed in the Traceability window, and you can follow the threads starting at each element through the diagram.</p> <p>If you require a rapid, broad-brush view of relationship flows in the project structure, starting with a general list of - say - all functional Requirements, you can use a combination of Model Search, Project Browser and Traceability window; this is a powerful tool for scanning your project, identifying how elements have been organized, and how they interact.</p> <p>For example: the Model Search would list all the requirements; you could rapidly click on each element and immediately see in the Project Browser where it has been grouped, and at the same time - in the Traceability window - how that element interacts with other elements in the model.</p> <p>By moving the cursor around a diagram or the Project Browser, and/or changing the relationship type combinations in the Traceability window, you can quickly see how elements are connected and how they influence each other.</p> <p>For example, you can see that - say - REQ017 is realized by two Use Cases, so you might then explore what else influences and is influenced by these two Use Cases; the Traceability window takes you well beyond what is likely to be depicted on any single diagram.</p> <p>If you have used transformations to develop your model, the T icon displays the transformation dependencies that exist between an element in a PIM and elements in the PSMs.</p>	<p>Example diagram ^[649]</p> <p>The Traceability Window ^[637]</p> <p>Model Search ^[611]</p> <p>The Project Browser ^[567]</p>
<p>Relationship Matrix</p>	<p>Using the Relationship Matrix, you can both create and study the relationships between, for example, the Requirements and Use Cases for a module.</p> <p>You might identify the 'theme' package (for example, <i>Manage Users</i>) in the Requirements model and in the Use Case model as the source and target packages, and explore the likely element and connector types in the packages; this, like the Traceability diagram, identifies which Requirements are (or should be) realized by which Use Cases.</p> <p>You can then perform similar checks with the <i>Manage Users</i> packages in, say, the Use Case and Implementation models.</p>	<p>Relationship Matrix ^[639]</p>

Topic	Detail	See also
	The Source and Target field browsers (...) enable you to examine child packages within the 'theme' package, and obtain further detail on how the feature at this stage is defined.	
Relationships Window	The Relationships Window shows a single level of relationships from the currently selected element. Each line corresponds to a single relationship and provides information about that relationship and the element at the other end of the relationship.	The Relationships Window ^[648]
Dependency Report	The Dependency report provides a list of dependencies for all elements in the model.	Dependency Report ^[2339]
Implementation Report	The Implementation report provides a list of all elements in the model that have to be implemented, and the elements that implement them.	Implementation Report ^[2341]

4.3.2 The Traceability Window

The Traceability window enables you to quickly see how elements are connected and how they influence each other.

- When an element is selected, it immediately becomes the top point in the Traceability window
- When a diagram is selected, all elements in the diagram are listed in the Traceability window, enabling you to use any of them as a starting point.

Access **View | Traceability (Ctrl+Shift+4)**

Use to

- Locate related elements in the Project Browser
- View the properties of related elements
- Show the diagrams where related elements are used
- Add related elements to the current diagram
- View the source code for related elements
- Change the display using the window toolbar options below:

Option	Action	See also
Generalizations	Show Generalization connectors	
Aggregations	Show Aggregation and Composition connectors	

Option	Action	See also
Nesting	Show Element nesting and Nesting connectors	
Realizations	Show Realization connectors	
Dependencies	Show Dependency connectors	
Transitions	Show State Transition, Control Flow, and Object Flow connectors	
Other Links	Show the following connectors: <ul style="list-style-type: none"> • Association • Use Case • Delegate • Assembly • Deployment • Information Flow • Manifest 	
Classifiers	Show where an element is used as the classifier of another element	
Embedded Element Reuse	Show where a Port or Part is also represented in another part of the model	
Transformations	Show where an element is created by running a Transformation from another element	Model Transformation ¹⁶⁷⁹
Custom References	Show the Custom References that have been added between elements	Set Up Cross References ⁸⁰³
Qualified Names	Show element names including the names of owning objects	

Notes

- Restricting the relationship types shown to the smallest set you are interested in makes it easier and faster to find the elements you are interested in
- As an alternative view or to include the information in documentation, the Traceability window aids you

in creating a Traceability diagram

Learn more

- [Create Traceability Diagrams](#) ^[649]

4.3.3 Relationship Matrix

The Relationship Matrix is a spreadsheet display of relationships between model elements within packages.

Topic	Detail	See also
Overview	<p>The Relationship Matrix is a convenient method of visualizing relationships quickly and definitively.</p> <p>It also enables you to create, modify and delete relationships between elements with a single mouse click - another quick way to set up complex sets of element relationships with a minimum of effort.</p> <p>On the Relationship Matrix, you select:</p> <ul style="list-style-type: none"> • A source package • A target package • The relationship type and • The relationship direction <p>Enterprise Architect identifies all the relationships between source and target elements by:</p> <ul style="list-style-type: none"> • Listing the source package elements down the side of the matrix • Listing the target package elements across the top of the matrix and • If a relationship exists between a source and target element, highlighting the intersecting grid square and displaying an arrow indicating the direction of the relationship <p>The direction is a reflection of which elements are the source elements and which the target (it does <i>not</i> indicate the Direction property of the connector, as defined in the connector Properties dialog); the display might also show a 'bent arrow' icon, indicating that connectors exist in both directions between the source and target elements.</p> <p>The highlighted squares are normally green, indicating that the <i>source</i> element is not locked (because the parent package has not been checked in under version control); if the element <i>is</i> locked (the parent package has been checked in) the highlight is pink.</p> <p>If you click on any square in the matrix, the square, the row header and the column header are highlighted.</p>	<p>Include Other Users' Packages ^[370]</p>
Operations	<p>Once you open the Relationship matrix, you can:</p> <ul style="list-style-type: none"> • Update, delete and create relationships through the Relationship 	<p>Open the Relationship Matrix ^[640]</p>

Topic	Detail	See also
	<p>Matrix</p> <ul style="list-style-type: none"> • Select options for modifying the type of information the Relationship Matrix displays • Export the contents of the Relationship Matrix to a CSV file, a .png file or a .emf file • Print the contents of the Relationship Matrix, scaled down if required • Save a profile of the Relationship Matrix settings to monitor development of the same source and target packages • Investigate the Source and Target elements in the relationship 	<p>Creating and Deleting Relationships ^[644]</p> <p>Relationship Matrix Options ^[645]</p> <p>Matrix Profiles ^[643]</p> <p>Review Source and Target Elements ^[647]</p>

4.3.3.1 Open the Relationship Matrix

Access **View | Relationship Matrix**
Project Browser Package Context Menu | Documentation | Open in Relationship Matrix | As Source or **| As Target**

Once the Relationship Matrix opens you:

- [Select which element type to show](#) ^[640]
- [Select the connector type and direction to show](#) ^[641]
- [Set the source and target packages](#) ^[641]

The Relationship Matrix refreshes after every change you make to the input parameters.

4.3.3.2 Set Element Type

The Relationship Matrix defaults to show all element types, or you can specify which types to show.

How to

To set the source and/or target element types

Step	Action	See also
1	Click on the drop-down arrow in the Type field for the source package, and/or for the target package.	
2	Find the required element type in the list and click on it. Enterprise Architect refreshes the Relationship Matrix content.	

4.3.3.3 Set Connector Type and Direction

In the Relationship Matrix, you must set the connector type to report on and the connector direction.

How to

To set the connector type and direction

Step	Action	See also
1	Click on the drop-down arrow in the Link Type field.	
2	Find the required connector type in the list and click on it.	
3	Click on the drop-down arrow in the Direction field.	
4	<p>Find the required direction in the list and click on it.</p> <p>Enterprise Architect refreshes the Relationship Matrix content.</p> <ul style="list-style-type: none"> If you set Direction to Both, each relationship is indicated by two arrows - a <i>From-To</i> arrow and a <i>To-From</i> arrow The direction is a reflection of which elements are the source elements and which are the target.; it does not indicate the Direction property of the connector, as defined in the connector Properties dialog 	

4.3.3.4 Set Source and Target Package

You must set both the source and target packages for the Relationship Matrix before relationships can be displayed.


You set the source and target packages AFTER setting the connector and element types/details; as Enterprise Architect refreshes the content after each change, this is usually faster.

The Relationship Matrix includes ALL child elements in a hierarchy. Sometimes in a large model this can be a lot of elements, possibly too many to be useful. Take care in selecting the source and target package.

How to

To set the source and target packages

Step	Action	See also
1	In the Project Browser, click on the required source package, then press (Ctrl) and click on the required target package, to select the two packages together.	

Step	Action	See also
2	<p>Drag the selected packages over the Source and Target fields.</p> <p>The first-selected package name displays in the Source field, and the second-selected package name displays in the Target field.</p>	
<i>Alternatively</i>		
1	<p>Select and drag a single package name over the Source OR Target field, to change just the source or the target package.</p> <p>If you drop the package name anywhere else on the Relationship Matrix, the system prompts you to specify whether to add it to the Source or Target field.</p>	
<i>Alternatively</i>		
1	<p>Click on the (...) (Browse) button at the end of the Source or Target field.</p> <p>A short menu displays, with options to:</p> <ul style="list-style-type: none"> • Locate a package using the Browse Project dialog or • Run a Model Search (selected from a drop-down list) on a search term; the name of the search is displayed in the Source or Target field <p>The target of the search depends on whether the Return matching items for the selected Package option is selected in the Search definition; if it is selected, the search operates on the current package, otherwise the search operates on the whole model</p> <p>If required, you can browse for a package in one field and run a search in the other.</p>	<p>Advanced Search Options </p>
2	<p>Select the required package, or the required Model Search and search term, and click on the OK button.</p>	

In all cases, Enterprise Architect immediately:

- Populates the Relationship Matrix axes with the elements identified in the Source and Target packages or the searches that meet the selection criteria, and
- Shows any relationships between the sets of elements that also meet the selection criteria

4.3.3.5 Matrix Profiles

If you use a particular combination of field values often, you can save that combination as a profile to make it easier to recall them for later use.

How to

To save a certain Relationship Matrix configuration as a named profile

Step	Action	See also
1	Set up the Relationship Matrix with the required: <ul style="list-style-type: none"> • Source and target packages or Model Searches • Source and target element types and • Connector type and direction 	Set Element Type ^[640] Set Connector Type and Direction ^[641] Set Source and Target Package ^[641]
2	Click on the Options button on the Relationship Matrix and select the Profiles Save as New Profile menu option. A short dialog displays.	
3	In the Enter Value field, type a profile name of up to 12 characters. Click on the OK button.	
4	Once you have created a profile, you can select it by clicking on the Profile drop-down arrow at the top of the Relationship Matrix screen and selecting the profile name from the list. Profiles are also listed in the <i>Matrix Profiles</i> folder of the Resources window; double-click on a profile in the folder to display the Relationship Matrix with that profile active.	Resources ^[1003]

Notes

- You can modify an existing profile by:
 - Selecting the profile name from the **Profile** drop-down list
 - Selecting the **Options: Profiles | Update Current Profile** menu option
 - Changing the field values, then
 - Saving the profile under the same profile name
- To delete an existing profile, select it in the **Profile** drop-down list and select the **Options: Profiles | Delete Current** menu option
- You can capture Relationship Matrix profiles in RTF reports

Learn more

- [Reporting Profiled Relationship Matrices](#)^[2310]

4.3.3.6 Creating and Deleting Relationships

From the Relationship Matrix, you can:

- Modify relationships
- Delete relationships
- Create new relationships

Use to

- Quickly create and manage relationships like Realization and Aggregation between Requirements and implementation elements (such as Use Cases)

How to

To modify or delete a relationship on the Relationship Matrix

Step	Action	See also
1	<p>Right-click the required highlighted relationship square on the matrix, and select the appropriate context menu option:</p> <ul style="list-style-type: none"> • View relationship - opens the Properties dialog for the selected relationship • Source element properties - opens the Properties dialog for the source element • Target element properties - opens the Properties dialog for the target element • Delete relationship 	
2	<p>If you have selected Delete relationship, Enterprise Architect prompts you to confirm this action.</p> <p>Click on the Yes button.</p> <p>The Delete relationship option is not available if:</p> <ul style="list-style-type: none"> • The source element (that is, the owner) is locked • You have selected Both in the Direction field - you are effectively trying to delete half a relationship 	
3	<p>If you have selected one of the other options, modify the properties in the dialog fields as required.</p> <p>Click on the OK button to save the changes.</p>	

To create a new relationship

Step	Action	See also
1	In the Link Type field, select the required relationship type.	
2	Right-click on the empty square at the intersection of the appropriate source row and target column.	
3	Select the Create new relationship context menu option. A submenu displays, listing the types of relationship you can create.	
4	Click on the required type of relationship; a connector of that type is created between the two elements.	

4.3.3.7 Relationship Matrix Options

The Relationship Matrix provides a number of menu options for modifying the output of information, for preserving the settings for such modifications, and for capturing the output.

Access [View | Relationship Matrix: Options](#)

Use to

- Output the information on the Relationship Matrix to the printer or to a metafile, .png file or .csv file
- Create and update profiles of the configurations of the matrix that you have designed
- Define local settings to control what the Relationship Matrix displays

Topics

Topic	Detail	See also
Print Relationship Matrix	Click on the Matrix Print menu option. The Print dialog displays, on which you select the output printer and specify the printer properties, the range of pages to print, and the number of copies. The output is a WYSIWYG representation of the Relationship Matrix contents.	
Display a preview of printout	Click on the Matrix Print Preview menu option. The Print Preview screen displays, showing the Relationship Matrix printout.	Print Preview 690

Topic	Detail	See also
<p>Scale printout</p>	<p>When you print the Relationship Matrix, by default it prints on as many pages wide and long as the matrix requires.</p> <p>You can scale the printout into a fixed number of pages wide, and the row height is automatically adjusted to maintain the proportions of the matrix; this reduces the overall size of the printout and improves appearance, especially when used in conjunction with the Landscape option in the printer properties.</p> <p>For example, a 16-page printout without scaling can, with a scaling of 2 pages wide, be reduced to 6 pages.</p> <p>To set the page scaling:</p> <ol style="list-style-type: none"> 1. Click on the Matrix Scale Setting menu option. The Scale Matrix dialog displays. 2. Select the Scale Matrix Width Into Pages checkbox, and type or select the number of pages in width to scale to 3. Click on the OK button to apply the setting. 	
<p>Save Relationship Matrix as graphic file</p>	<p>Click on the Matrix Save as Metafile or Matrix Save as PNG menu options.</p> <p>A Browser dialog displays that enables you to select the target file location and specify the filename of the .emf or .png file in which to save the output.</p> <p>You can incorporate these files in an RTF or HTML report, as a hyperlinked file or an included file.</p>	
<p>Export output to CSV file</p>	<p>The contents of the Relationship Matrix can be exported to a CSV file, which provides a convenient mechanism for moving the matrix data to a spreadsheet environment such as Microsoft Excel.</p> <p>This option is also active in the 'Lite', read-only version of Enterprise Architect.</p> <p>To export to CSV:</p> <ol style="list-style-type: none"> 1. Select the Matrix Export to CSV menu option. The Windows Browser dialog displays. 2. Browse to the required file location and type in a .CSV filename to export to. 3. Click on the Save button to export the data. 	<p>The Read-only 'Lite' Edition [26]</p>
<p>Create and manage matrix profiles</p>	<p>If you use a particular combination of field values often, you can save that combination as a profile to make it easier to recall them for use.</p>	<p>Matrix Profiles [643]</p>
<p>Manage display content</p>	<p>Select the Options menu option.</p> <p>The Matrix Options dialog displays; select one or more of the following checkboxes as required:</p> <ul style="list-style-type: none"> • Include Source Children - to recursively include child packages and contents under the Source • Include Target Children - to recursively include child packages and 	

Topic	Detail	See also
	<p>contents under the Target</p> <ul style="list-style-type: none"> • Include All Extended Meta Types - to include elements that are extensions of a specified meta-type; for example, if there are Block elements (extending Class) in the package, selecting this option and specifying the type Class includes Class and Block elements, and any further derivatives of Block in the matrix • Sort Axes - to ensure package elements display in alphabetical order • Show Package Names - to hide or show package names in the Relationship Matrix; this is useful for shortening the displayed texts, especially in circumstances where packages have long names • Use Element Alias If Available - to display an element's alias instead of the element name, if an alias has been defined • Show Level Numbering If Available - to reproduce level numbering in the Relationship Matrix, if it is turned on in the Project Browser 	

4.3.3.8 Review Source and Target Elements

As well as providing information on connectors and relationships, the Relationship Matrix enables you to obtain information on the source and target elements in a relationship.

Topic	Detail	See also
Identify elements in relationships with a source or target element	<p>Click on the source or target element name in the row or column titles.</p> <p>The entire row or column is highlighted.</p> <p>Scroll across or down the highlighted row or column and quickly identify where the relationships are; this is very useful if the row or column is long.</p>	
Display the Properties dialog for the selected element	<p>Right-click on the element name and select the Properties context menu option.</p>	
Identify diagrams in which the element is used	<p>Right-click on the element name and select the Find in Diagrams context menu option.</p> <p>Either:</p> <ul style="list-style-type: none"> • The only diagram in which the element is used displays, with the element highlighted, or • A list of the diagrams in which the element is used displays; you then double-click on the required diagram to open it 	

Topic	Detail	See also
Highlight the element name in the Project Browser	Right-click on the element name and select the Locate in Project Browser option. The Project Browser expands to the location of the element, and the element name is highlighted.	
Make the element the focus in any docked screens or windows that are open	Right-click on the element name and select the Set Context Item option; the selected element becomes the subject of any open windows or screens.	

4.3.4 The Relationships Window

The Relationships window displays all connectors between an element selected from a diagram or the Project Browser, and other elements. This provides a quick overview of an element's relationships in the model.

Access [View | Relationships](#)

Topics

Topic	Detail	See also
Content	<p>The Relationships window shows, for each connector, the:</p> <ul style="list-style-type: none"> • Connector type • Any stereotype on the connector • The Source and Target elements in the relationship • The direction of the relationship • Any stereotype on the Target element • The Target element type • The Target and Source element roles, if any <p>It also has a Target in Diagram column which, if the Target element for a connector is visible in the currently loaded diagram, contains the value Yes; this is very useful when you are dragging related elements from the Relationships list onto the current diagram.</p> <p>Adding the target element to the diagram does not change the context to that element; the originally-selected element (the subject of the Relationships window) remains in context, and you continue working with that element.</p>	
Facilities	<p>Double-click on a connector in the list to display the <connector type> Properties dialog, on which you can edit the connector attributes.</p> <p>Right-click on a connector to open the context menu, which provides options to:</p>	

Topic	Detail	See also
	<ul style="list-style-type: none">• Locate the related element in the Project Browser• View the related element properties• Delete the connector• Hide certain connectors in diagrams• Show hidden connectors• Locate the selected element in any other diagrams in which it occurs• Place the related element in the current diagram (if it is not currently visible in the diagram); this is useful in building a picture of what an element interacts with, especially when reverse engineering an existing code base	

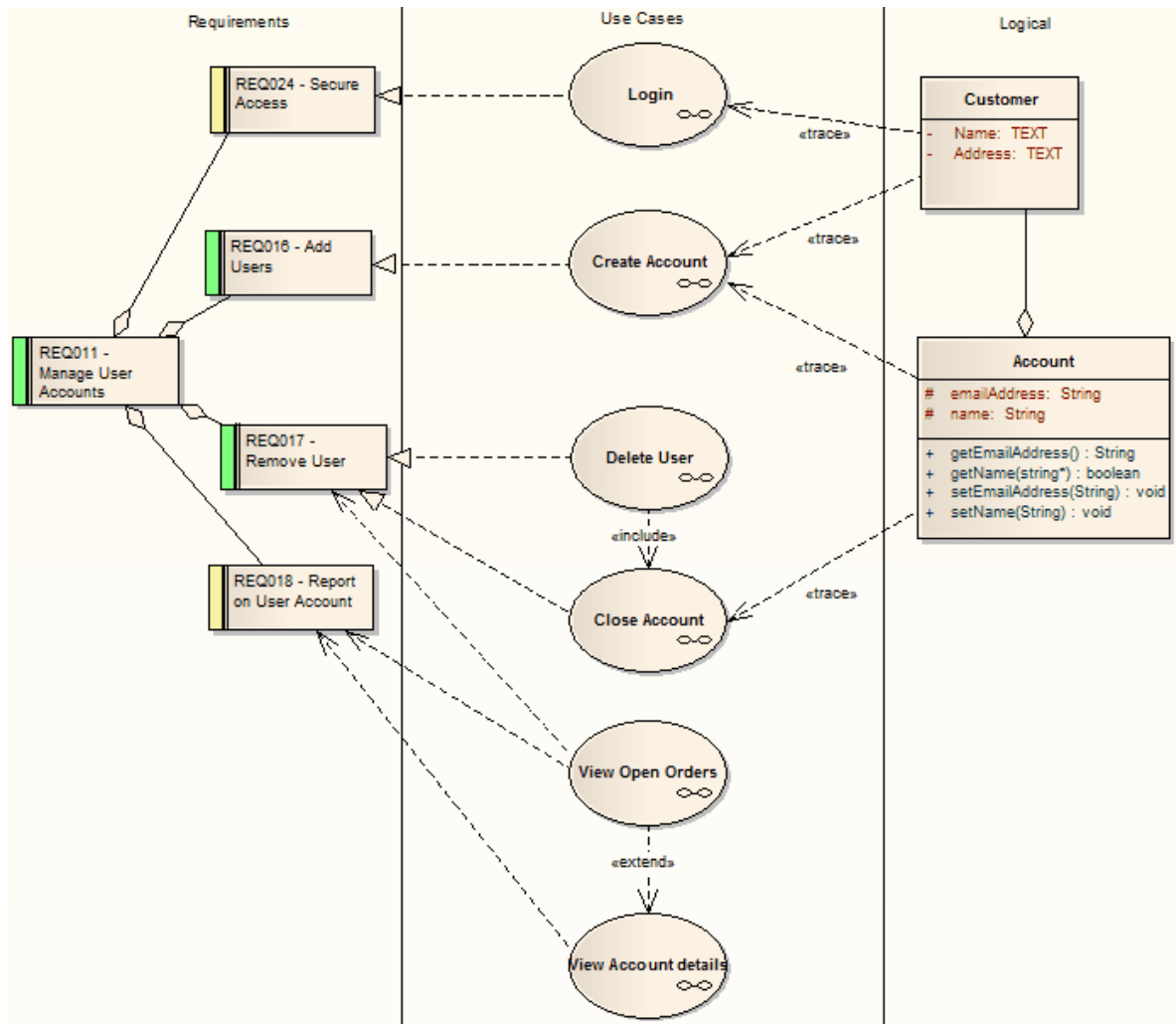
Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, with security on, the diagram and the source and target elements must be free for editing before some of these options are available for use

4.3.5 Example Diagram

A wide variety of relationships can be shown on diagrams; you are not restricted to having a single element type or elements from a single phase on a diagram. This can help you to create appropriate relationships (such as Realize or Trace) to elements where no relationship was previously defined.

Consider the following diagram:



Topic	Detail	See also
<p>Diagram Contents</p>	<p>The example diagram above shows how a series of requirements are traced to Use Cases and then Classes.</p> <p>The diagram instantly shows:</p> <ul style="list-style-type: none"> • That two levels of Requirements are realized by Use Cases • Which Requirement is realized by which Use Case(s) • How some of the Use Cases are implemented by Class elements <p>You can drill down on each Use Case (or, in other Traceability diagrams, any other composite element) to display more detailed diagrams showing how the Use Case meets the Requirement; the Close Account Use Case, for example, contains a Communication diagram and a Sequence diagram.</p> <p>You can tailor your Traceability diagrams to depict any level of granularity and any stages of development that are appropriate; for example, you might:</p> <ul style="list-style-type: none"> • Narrow the above diagram to show development from just the 	

Topic	Detail	See also
	<p>Remove User Requirement, or</p> <ul style="list-style-type: none"> Extend it to include Interfaces, Components, Test Case elements or any other facet of the system or process 	
Building a Diagram	<p>A number of tools are available to for creating diagrams that show traceability:</p> <ul style="list-style-type: none"> The diagram context menu Add Related Elements option enables you to automatically bring in elements linked to the selected element The Relationships window enables you to add elements to the diagram that are directly linked to the currently selected element The Traceability window enables you to add elements that are indirectly linked 	Insert Related Elements ^[833]

Notes

- Whilst the Traceability diagram itself provides information on the definition, design and implementation of a business process feature, much more information can be obtained using the Traceability tools

Learn more

- [Realize](#) ^[1254]
- [Trace](#) ^[1260]
- [Traceability Tools](#) ^[636]
- [The Relationships Window](#) ^[648]
- [The Traceability Window](#) ^[637]

4.3.6 Gap Analysis Matrix

The Gap Analysis Matrix is a tool to analyze model artifacts for potential gaps in solutions.

The idea is to highlight a shortfall between the Baseline Architecture and the Target Architecture; that is, items that have been deliberately omitted, accidentally left out, or not yet defined.

Access [View | Gap Analysis Matrix](#)

Topic	Detail	See also
Overview	<p>The Gap Analysis Matrix is a convenient method of identifying gaps between solution architectures.</p> <p>It enables you to model the gaps in the repository, which can later be addressed and assigned as tasks; the identified gaps can also be used to prioritize activities.</p> <p>On the Gap Analysis Matrix, you select the:</p>	Include Other Users' Packages ^[370]

Topic	Detail	See also
	<ul style="list-style-type: none"> • Target Architecture package • Baseline Architecture package • Type of Target Architecture artifact • Type of Baseline Architecture artifact • Type of element to model the gap 	
Operations	<p>Once you open the Gap Analysis Matrix, you can:</p> <ul style="list-style-type: none"> • Create, update and delete Gap Matrix Profiles to monitor development of the Baseline and Target packages • Edit Gap notes during a discussion session for Gap analysis • Create elements to address gaps • Add links to existing elements to address gaps • Remove links to existing elements modeled as gap 	Edit Gap Notes ⁶⁵⁴
Select Package	<p>Click on the (...) (Browse) button at the end of the Target Architecture or Baseline Architecture field.</p> <p>The Browse Project dialog displays.</p> <p>Select the required package and click on the OK button; by default all the elements under the package are displayed.</p>	
Set Element Type	<p>Click on the drop-down arrow in the Filter field for the Target Architecture package, and/or for the Baseline Architecture.</p> <p>Find the required element type in the list and click on it; Enterprise Architect refreshes the matrix content.</p>	
Set Gap Type	<p>Click on the drop-down arrow in the Record Gap As field.</p> <p>Find the required element type in the list and click on it.</p> <p>When you create a gap, the specified type of element is created; for example, if you want to address the gap with an issue element then select Issue for this field.</p>	
Baseline Architecture	<p>The Baseline Architecture artifacts are listed along the rows of the matrix.</p> <p>The last row is for capturing the new elements in the target architecture along the columns but not in the baseline architecture, hence the name New.</p>	
Target Architecture	<p>The Target Architecture artifacts are listed along the columns of the matrix.</p> <p>The last column is for capturing the missing or intentionally eliminated</p>	

Topic	Detail	See also
	elements in the baseline architecture along the rows but not in the target architecture; hence the name Missing / Eliminated .	
Modeling Gap	<p>Right-click on a cell in the New row or Missing / Eliminated column.</p> <p>The context menu displays options to:</p> <ul style="list-style-type: none"> • Create a Gap element • Add a link to an existing Gap element <p>If a link to Gap element already exists in the cell then the following options are displayed:</p> <ul style="list-style-type: none"> • Edit Gap element • Find in Project Browser... • Remove Gap element link 	
Create Gap	<p>Right-click on the cell and select the Create Gap Element context menu option.</p> <p>The Browse Project dialog displays.</p> <p>Select the package in which to create the Gap element and click on the OK button; a Gap element is created in the selected package and its Properties dialog displays to enable you to enter the element name and other required properties.</p>	
Link to Existing Gap	<p>If you intend to use a Gap element that is already available in the model, right-click on the appropriate cell in the Missing / Eliminated column or New row and select the Link to Existing Gap Element context menu option.</p> <p>The Select Classifier dialog displays, enabling you to select the existing Gap element.</p>	
Remove Link to Gap	<p>If you intend to remove a link to the Gap element in a cell, right-click on the appropriate cell in the Missing / Eliminated column or New row and select the Remove Link to Gap Element context menu option.</p> <p>The link is removed from the cell but the element still exists in the Project Browser.</p>	
Review Gap Element	<p>Right-click on the appropriate cell in the Missing / Eliminated column or New row and select the Edit Gap Element context menu option.</p> <p>The Properties dialog displays, enabling you to edit the selected Gap element.</p> <p>To locate the element in the Project Browser select the Find in Project Browser... context menu option, which highlights the element in the Project Browser.</p>	

Topic	Detail	See also
Gap Analysis Matrix Profiles	<p>On the Gap Analysis Matrix, you can create and manage profiles to save commonly-used combinations of Target Architectures, Baseline Architectures and stereotypes.</p> <p>To work on Gap Analysis Matrix profiles, click on the Options button in the top right corner of the matrix; a submenu displays, listing options to:</p> <ul style="list-style-type: none"> • Create / Save the current matrix settings and content • Update the currently selected profile in the Profile field • Delete the currently selected profile in the Profile field 	
Open a saved Profile	<p>The Profile field drop-down list shows all the saved Gap Analysis Matrix profiles.</p> <p>Click on the drop-down arrow in the Profile field, find the required Profile in the list and click on it to load the content of the selected profile in the matrix.</p>	

Notes

- The Gap Analysis Matrix is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect

4.3.6.1 Edit Gap Notes

The Gap Analysis Matrix provides a way to record notes during the initial phase or during a discussion session for Gap Analysis. These notes are saved in profiles, which can later be reviewed and the appropriate model elements created to address these gaps.

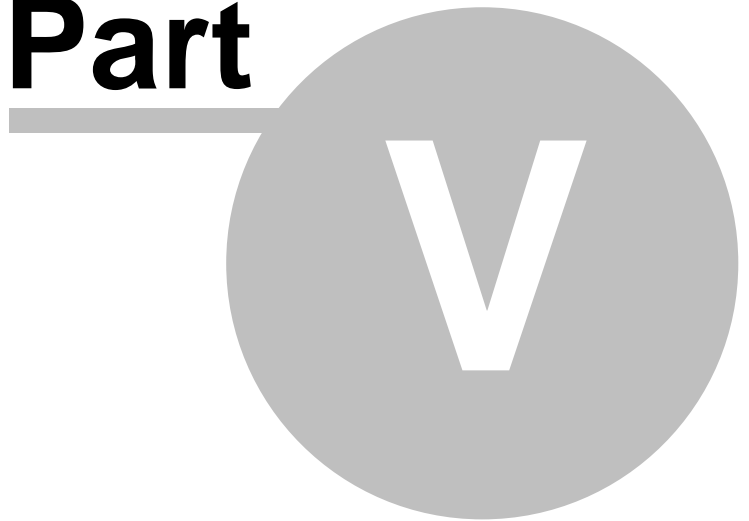
How to

To edit gap notes

Step	Action	See also
1	<p>Right click on the cell at the intersection of a Target and Baseline element and select the Edit Gap Note context menu option.</p> <p>Alternatively, double click on the cell.</p> <p>The Gap Note dialog displays.</p>	Gap Analysis Matrix ^{65↑}
2	Edit the notes as required and click on the OK button.	
3	The edited notes display in the selected cell in the matrix.	
4	Update or save the matrix to retrieve it when the Gap Matrix profile is loaded at a	

Step	Action	See also
	later stage.	

Part



5 Modeling Basics



Modeling can be described as graphically representing a business process or software system. The resulting model can be used to emphasize a certain aspect of the system being represented, and to record, document and communicate its detail. A study of such a model can enable insight or understanding of the system. Enterprise Architect's modeling platform is based on the Unified Modeling Language (UML), a standard that defines rules and notations for specifying business and software systems.

Using Enterprise Architect, you can quickly build a model using a hierarchy of *packages* to represent the structure and organization of the model.

Topics

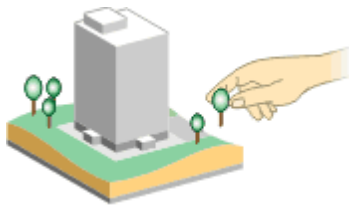
Topic	Link
<p>Models - a model is the highest conceptual level, representing a distinct and complete representation of all or some part of a modeled system.</p> <p>A Project can contain multiple models.</p>	<p>Models ⁶⁶⁰</p> <p>Modeling ⁶⁵⁹</p> <p>Model Wizard ⁶⁶⁰</p>
<p>Views are the second level within a model and define a specific viewpoint of the system being modeled - for example a Use Case view, a Requirements View or a Dynamic (behavioral) View.</p> <p>Views are simply packages which have an additional conceptual meaning.</p>	<p>Views ⁶⁷⁶</p> <p>Add Views ⁶⁷⁷</p>
<p>Packages are the basic containers that create the overall model structure. Packages hold other packages, elements, diagrams and similar model constructs.</p>	<p>Packages ⁶⁸⁰</p> <p>Open Package in the Project Browser ⁶⁸⁰</p>
<p>Diagrams are visual representations of how model elements are connected or behaviorally related. They can also display the characteristics of an element, such as attributes, methods, notes and tagged values in a convenient visual style.</p>	<p>Diagrams ⁶⁸⁷</p> <p>Diagram Toolbox ⁶⁹⁸</p> <p>The Quick Linker ⁷⁹¹</p> <p>Diagram Context Menu ⁶⁸⁷</p> <p>Diagram Context Menu ⁶⁸⁷</p> <p>Diagram Tabs ⁶⁹⁵</p> <p>Diagram Tasks ⁷²¹</p> <p>Layout Diagrams ⁷⁶⁹</p>
<p>Elements are the basic building blocks of models. They represent both</p>	<p>Elements ⁷⁹⁵</p>

<p>structural constructs such as Classes and Interfaces, as well as behavioral constructs such as Activities, Actions and States.</p>	<p>Element Context Menu ^[827] Visual Representation ^[841]</p>
<p>Connectors are the various kinds of relationships between elements within a model - including behavioral relationships, associations, taxonomic relations and similar.</p>	<p>Connectors ^[936] Connector Context Menu ^[936] Connector Tasks ^[942] Connector Properties ^[961]</p>
<p>Tagged Values are customizable, profile or user defined properties which are generally related to a Stereotype and define additional attributes and characteristics of an element. They are one of the fundamental means of extending UML into more domain specific areas.</p>	<p>Tagged Values ^[969] Quick Add - Tagged Value To Elements ^[971] Modify Tagged Values ^[973]</p>
<p>Notes are the standard common language based descriptions of what an element, diagram, feature or relationship is for and how it is used within the model. Notes are often used as a first step to eliciting the meaning and use of an element - which is later refined into something more concrete and precisely specified.</p>	<p>Notes ^[977] Notes Toolbar ^[979]</p>
<p>Reference Information - a variety of basic types and information that is used across a particular model - for example, Stereotypes, Tagged Values and similar.</p>	<p>Reference Data ^[981] UML Types ^[981]</p>

Learn more

- For information on UML, see the [Standard UML Models](#) ^[1008] topic
- The [Quick Start Tutorial](#) ^[49] topic briefly shows you how to create a diagram within a package, containing elements and connectors
- Sparx Systems also provide a [Demonstration of quickly developing a Use Case model](#) (Online Resource)

5.1 Modeling



Enterprise Architect is a comprehensive UML analysis and design tool. It provides a library of UML data structures that you can use and extend to develop your models.

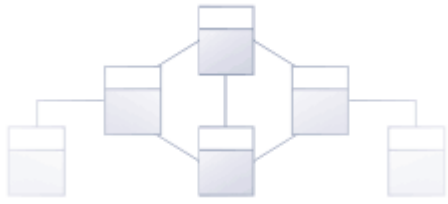
Topics

Topic	Detail	See also
UML Data Structures	<p>You create your projects and models using the Start Page or File menu, which provide templates on which to base your models.</p> <p>You initially create your packages and diagrams using the toolbars and menus, and the elements and connectors using the Diagram Toolbox.</p> <p>You can also create new structures through the Project Browser, and re-use existing structures using the Project Browser, Model Views, Package Browser and Model Search.</p> <p>Building models requires the use of various UML data structures and Enterprise Architect tools, as above, to graphically represent a business process or software system; the resulting model can be used to emphasize a certain aspect of the system being represented and to record and communicate its detail.</p>	<p>The Start Page ^[87]</p> <p>File Menu ^[92]</p> <p>Model Wizard ^[660]</p> <p>Workspace Toolbars ^[138]</p> <p>Main Menu ^[90]</p> <p>Diagram Toolbox ^[698]</p> <p>Project Browser ^[567]</p> <p>Model Views ^[598]</p> <p>Package Browser ^[587]</p> <p>Model Search ^[617]</p>
Relationship Matrix	<p>A further extremely useful tool is the Relationship Matrix, which enables you to visualize and amend the relationships and hence organization of structures within the model.</p> <p>Enterprise Architect also provides particular support for:</p> <ul style="list-style-type: none"> • Requirements Management and • Modeling the business process, an essential part of any software development process <p>You can extend the scope of your models using UML Stereotypes, Profiles and Patterns, and MDG Technologies.</p>	<p>Modeling Fundamentals ^[657]</p> <p>Build Your Own Modeling Language ^[1292]</p> <p>Requirements ^[1468]</p>

Learn more

- [Standard UML Models](#) ^[1008]

5.2 Models



In Enterprise Architect a *model* is a special type of package, being the top level entry point to an Enterprise Architect project file.

Topics

Topic	Detail	See also
Models	<p>You can develop a project with one model, or with several.</p> <p>Each model is a root node of a hierarchy of Model Packages and Views and, below them, packages that contain the diagrams, elements, relationships and associated metadata that define the structure and function of a system or process.</p> <p>These components are organized through the package hierarchy, which helps to group and manage related components; by iterating through all models, you can access all the elements within the project.</p> <p>You can create the model or models when you first create the project, or you can add and develop new models later.</p> <p>When you first create a project, Enterprise Architect automatically displays the Model Wizard, which you use to generate the required models and subordinate structures from a wide selection of Model Patterns.</p> <p>However, you can also create a model as a single project root node with no content, and build up the model structure yourself:</p> <p style="text-align: center;">Project Browser context menu Add Add Model (root node)</p> <p>You can also delete a model, but be aware that everything contained in the model is deleted as well.</p> <p style="text-align: center;">Right-click Delete Project Root</p>	<p>Modeling Basics ⁶⁵⁷</p> <p>Model Wizard ⁶⁶⁰</p>

5.2.1 Model Wizard

The Model Wizard enables you to create models in your project, based on Patterns from a range of technologies, both when you first create the project and over time.

The Model Wizard has three tabs:

- Model Patterns - these enable you to create basic models that you can develop further as components of your project
- Application Patterns - these enable you to generate starter projects including model information, code and build scripts for one of several basic application types
- VEA Examples - these enable you to import complete sample models (packages), including all

necessary model information, code and build scripts, to explore and try out the Visual Execution Analyzer

This topic discusses use of the Model Patterns tab. See *Learn more*, below, for links to topics on the other two tabs.

Access Automatically displays when creating a new project; otherwise
 Right-click on a Project Browser Package node | Add | Add a New Model using Wizard (Ctrl) + (Shift) + (M)

Project Browser toolbar | 

Use to

- Add new models to a project

Reference

Field	Usage	See also
Technology	<p>Lists the technologies integrated with or imported into Enterprise Architect, which provide Model Patterns specific to the technology.</p> <p>If you are a Technology Developer, you can create and import your own MDG Technology and distribute custom Model Patterns as part of that technology; the name of your technology displays in this list.</p> <p>Click on the required technology to list the associated Model Patterns in the Name panel.</p> <p>The first three technologies support core UML modeling:</p> <ul style="list-style-type: none"> • <i>Basic UML 2 Technology</i> - provides Model Patterns for general purpose, 'diagram type' UML modeling, generating extensive model structures • <i>Simple UML Views</i> - provides Model Patterns for 'view' oriented modeling, generating an appropriate package with an empty diagram, eliminating the need to delete unwanted sections of boilerplate structure <p>These are slightly different from the basic Views that you can create instead of Model Packages, under the model root node</p> <ul style="list-style-type: none"> • <i>Core Extensions</i> - provides Model Patterns for the extended application of UML modeling <p>There is some overlap between these three categories.</p>	<p>Views ⁶⁷⁶¹</p>

Field	Usage	See also
Name	List the Model Patterns provided by the selected technology. Click on the check box for each model type you want to create in your project.	
All	Select all of the Model Patterns listed in the Name panel.	
None	Clear all Model Patterns selected in the Name panel.	
OK	Generate the model structures for your project. The structures are added to the Project Browser and can be expanded and worked on immediately.	
Cancel	Abort the creation of model structures.	
Help	Display this Help topic.	

Learn more

- [Model Templates](#)^[662]
- [Application Patterns](#)^[1738]
- [Visual Execution Analyzer Samples](#)^[2145]
- [Custom Templates](#)^[1355]

5.2.2 Model Templates

Model templates generated by Enterprise Architect are designed to assist in the creation of projects and models for both new and experienced users.

The templates are provided through the Model Wizard, for the model types available under the Basic UML 2 Technology and Core Extensions technology.

Each template provides a framework of packages, diagrams, representative elements and connectors on which to build a model appropriate to your purpose. These include overview diagrams that provide an introduction to the terminology and icons used in the model templates, and give a quick guide to the UML concepts important to the templates and how they are applied in Enterprise Architect. This information is provided either in Note elements or via Hyperlinks to the Enterprise Architect Help and other Sparx Systems website resources.

Templates:

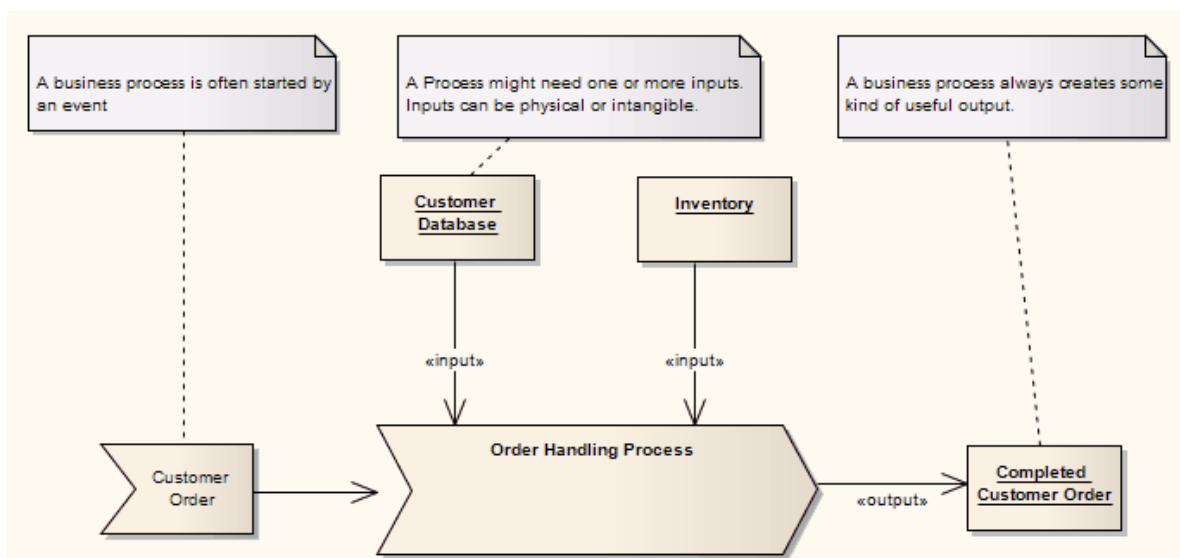
Template	Detail	See also
Business Process Model	<p>The Business Process model describes both the behavior and the information flows within an organization or system.</p> <p>As a model of business activity, it captures the significant events, inputs, resources, processing and outputs associated with relevant business processes.</p>	<p>The Business Process Model</p> <p>Business Models ^[1512]</p> <p>Analysis Diagram ^[1509]</p> <p>Business Modeling/Interaction ^[1514]</p>
Requirements Model	<p>The Requirements model is a structured catalogue of end-user requirements and the relationships between them.</p> <p>The Requirements Management facilities built into Enterprise Architect can be used to define Requirement elements, connect Requirements to other model elements, connect Requirements into a hierarchy and report on Requirements.</p>	<p>Requirements Management ^[1468]</p> <p>Requirements Management in Enterprise Architect Packages ^[680]</p>
Use Case Model	<p>The Use Case model describes a system's functionality in terms of Use Cases.</p> <p>Each Use Case represents a single repeatable interaction that a user or 'actor' experiences when using the system, emphasizing the user's perspective of the system and interactions.</p>	<p>Use Case ^[1174]</p> <p>Use Case Diagram ^[1030]</p> <p>The Use Case Model</p>
Domain Model	<p>A Domain model is a high-level conceptual model, defining physical and abstract objects in an area of interest to the Project.</p> <p>The Domain model can be used to document relationships between and responsibilities of conceptual classes (that is, classes that represent the concept of a group of things rather than Classes that define a programming object).</p> <p>This model is also useful for defining the terms of a domain.</p> <p>A Domain model shows:</p> <ul style="list-style-type: none"> • The physical and organizational units of the domain; for example, <i>Employee</i> and <i>Flight</i> • The relationships between these units; for example, Employee is <i>assigned to</i> Flight • The multiplicity of those relationships; for example, <i>one</i> employee can be assigned to <i>no</i> flights, <i>one</i> flight or <i>many</i> flights (represented by 1* at the ends of the relationship) 	<p>Multiplicity ^[965]</p>
Class Model	<p>The Class model is a rigorous, logical model of the software system under construction.</p> <p>Classes generally have a direct relationship to source code</p>	<p>Classes ^[1181]</p> <p>Class Diagram ^[1014]</p>

Template	Detail	See also
	or other software artifacts that can be grouped together into executable components.	The Logical Model
Database Model	The Database model describes the data that must be stored and retrieved as part of the overall system design. Typically, this means relational database models that describe the tables and data in detail and enable generation of DDL scripts to create and set up databases.	Data Models ^[1625] UML Database Modeling
Component Model	The Component model defines how Classes, Artifacts and other low level elements are collected into high level components, and describes the interfaces and connections between them. Components are compiled software artifacts that work together to provide the required behavior within the operating constraints defined in the Requirements model.	Components ^[1187] Component Diagram ^[1023] The Component Model
Deployment Model	The Deployment model describes how and where a system is to be deployed. Physical machines and processors are represented by Nodes, and the internal construction can be depicted by embedding Nodes or Artifacts. As Artifacts are allocated to Nodes to model the system's deployment and roll out, the allocation is guided by the use of deployment specifications.	Nodes ^[1196] Artifacts ^[1179] Deployment and Roll Out ^[81] Deployment Diagram ^[1020] Compartments - Tagged Values ^[842] The Physical Model
Testing Model	The Testing model describes and maintains a catalogue of tests, test plans and results that are executed against the current model.	Testing ^[2217] Test Case ^[1677] Show Test Script Compartments ^[2229] Testing Support in Enterprise Architect
Maintenance Model	The Maintenance model enables visual representation of issues arising during and after development of a software product. The model can be enhanced with the integration of change elements and testing.	Development ^[2236] Color Code External Requirements ^[1477]
Project Management Model	The Project Management model details the overall project plan, phases, milestones and resource requirements for the current project. Project Managers can use Enterprise Architect to assign resources to elements, measure risk and effort and to	Project Management ^[447] Project Manager (Online Resource)

Template	Detail	See also
	estimate project size. Change control and maintenance are also supported.	
User Interface Model	<p>The User Interface Model is a high level, logical mapping of forms, web pages, dialogs and other screens and controls that form part of the proposed system.</p> <p>The template does not automatically include web interface or WIN32 UI design elements, but you can incorporate these in your subsequent development.</p>	User Interface Models ^[1655] Web Stereotypes ^[1660] Win32 UI Technology ^[1661]

5.2.2.1 Business Process Model Template

The *Business Process model* describes both the behavior and the information flows within an organization or system. As a model of business activity, it captures the significant events, inputs, resources, processing and outputs associated with relevant business processes.



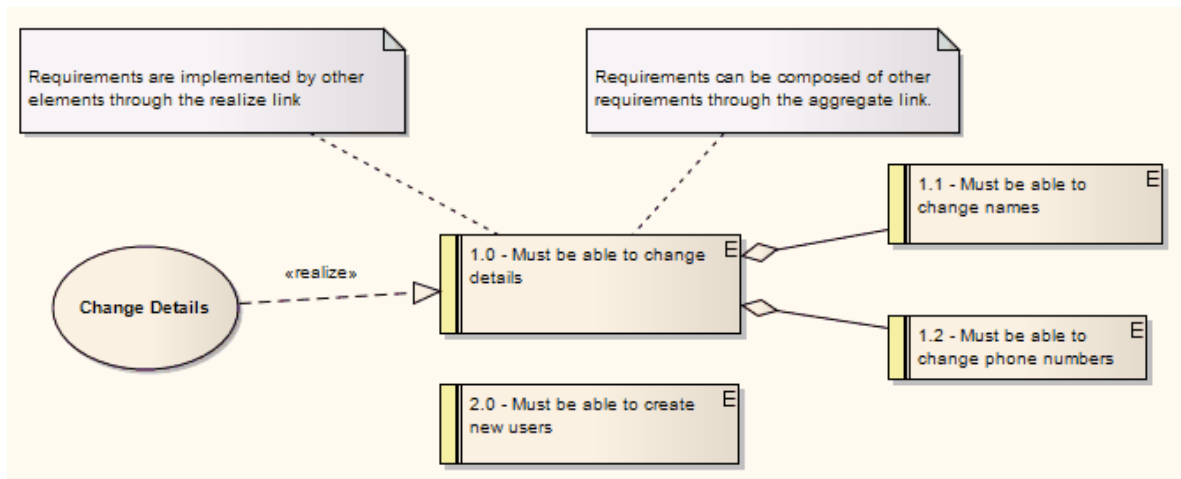
Learn more

- [The Business Process Model](#)
- [Business Models](#) ^[1512]
- [Analysis Diagram](#) ^[1509]
- [Business Modeling/Interaction](#) ^[1514]

5.2.2.2 Requirements Model Template

The *Requirements model* is a structured catalogue of end-user requirements and the relationships between them.

The Requirements Management facilities built into Enterprise Architect can be used to define Requirement elements, connect Requirements to other model elements, connect Requirements into a hierarchy and report on Requirements.



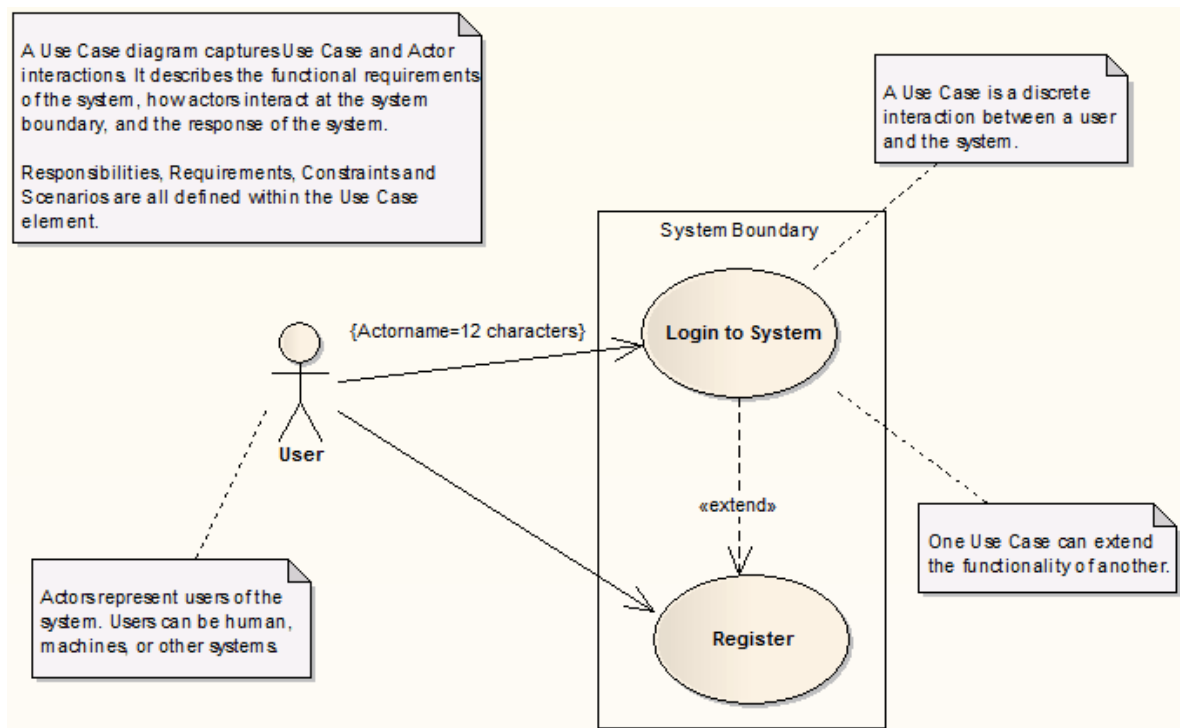
Learn more

- [Requirements Management](#)^[1468]
- [Requirements Management in Enterprise Architect](#)
- [Packages](#)^[680]

5.2.2.3 Use Case Model Template

The *Use Case model* describes a system's functionality in terms of Use Cases.

Each Use Case represents a single repeatable interaction that a user or 'actor' experiences when using the system, emphasizing the user's perspective of the system and interactions.



Learn more

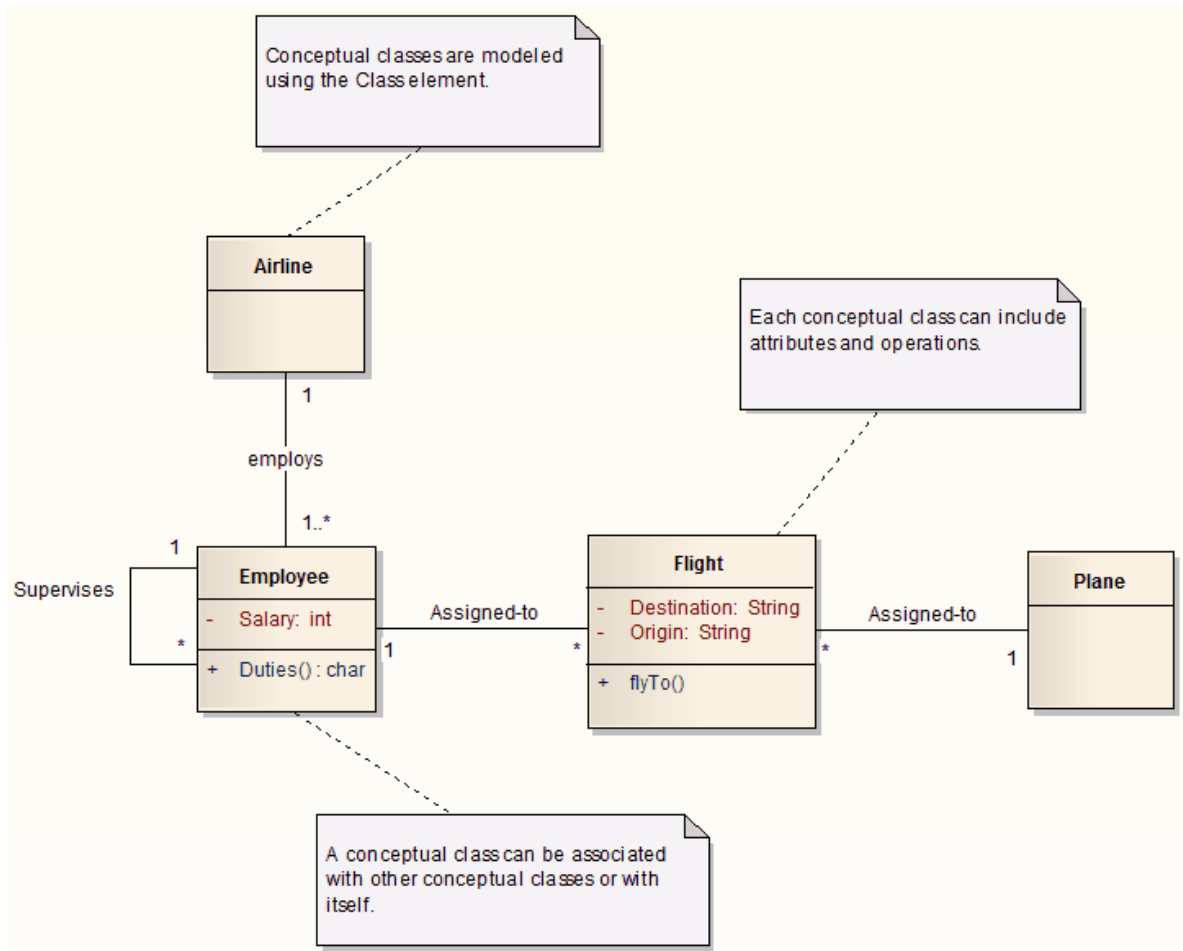
- [Use Case](#)^[1174]
- [Use Case Diagram](#)^[1030]
- [The Use Case Model](#)

5.2.2.4 Domain Model Template

A *Domain model* is a high-level conceptual model, defining physical and abstract objects in an area of interest to the Project.

The Domain model can be used to document relationships between and responsibilities of conceptual classes (that is, classes that represent the concept of a group of things rather than Classes that define a programming object).

It is also useful for defining the terms of a domain.



A Domain model shows:

- The physical and organizational units of the domain; for example, *Employee* and *Flight*
- The relationships between these units; for example, *Employee* is *assigned to* *Flight*
- The multiplicity of those relationships; for example, *one* employee can be assigned to *no* flights, *one* flight or *many* flights (represented by the 1 and the * at the ends of that relationship)

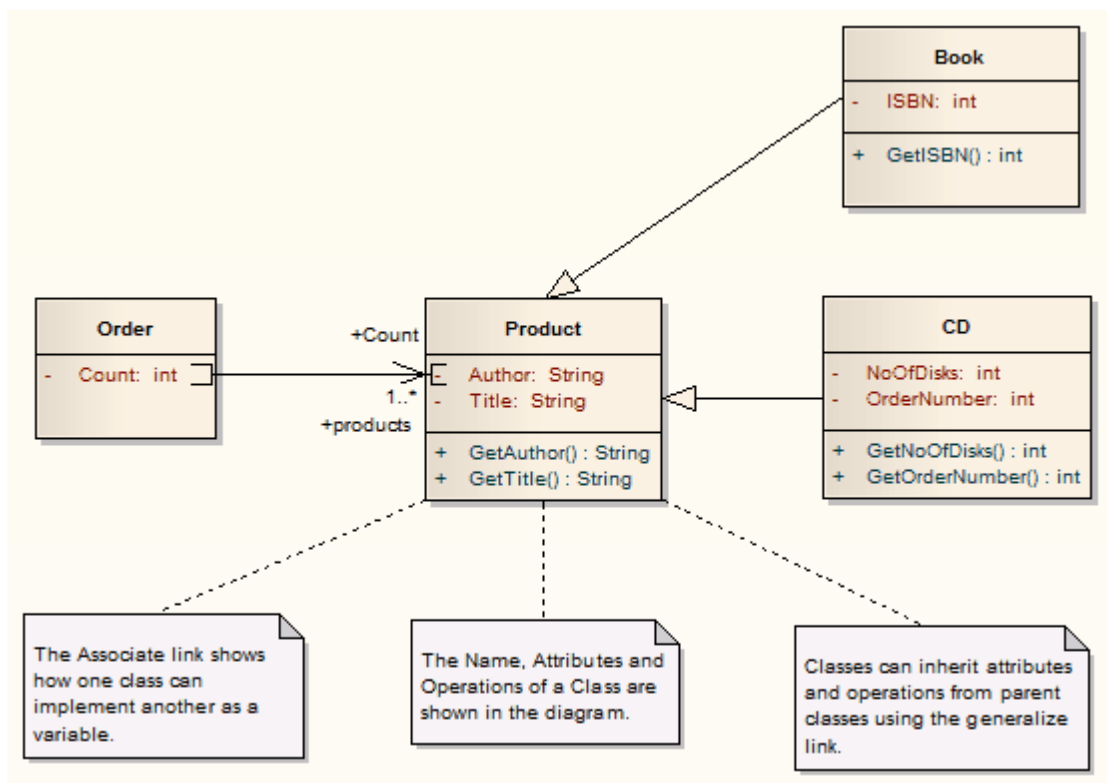
Learn more

- [Multiplicity](#)^[965]

5.2.2.5 Class Model Template

The *Class model* is a rigorous, logical model of the software system under construction.

Classes generally have a direct relationship to source code or other software artifacts that can be grouped together into executable components.

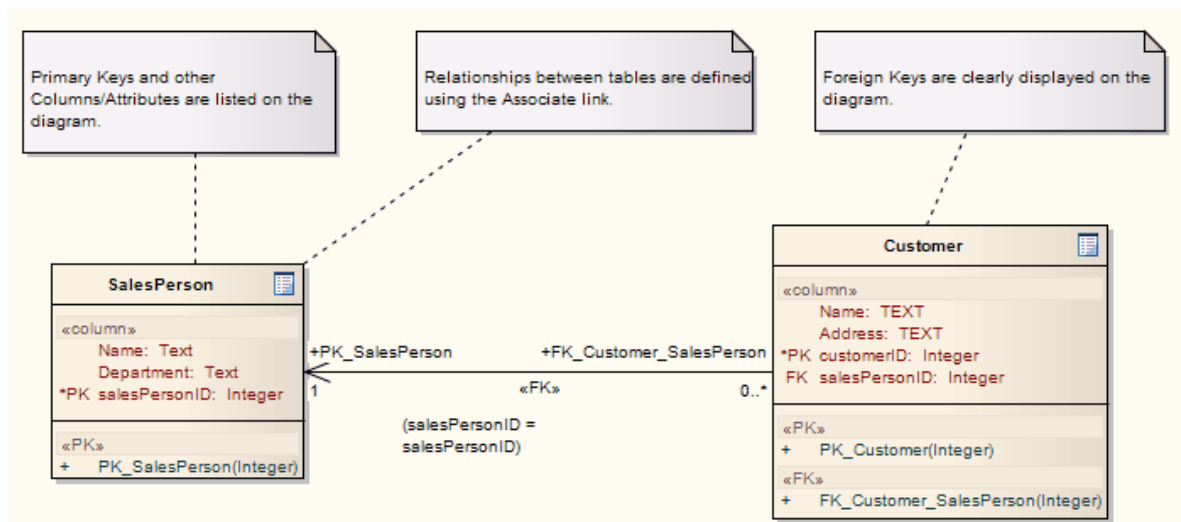


Learn more

- [Classes](#) ^[118]
- [Class Diagram](#) ^[1014]
- [The Logical Model](#)

5.2.2.6 Database Model Template

The *Database model* describes the data that must be stored and retrieved as part of the overall system design. Typically this means relational database models that describe the tables and data in detail and enable generation of DDL scripts to create and set up databases.



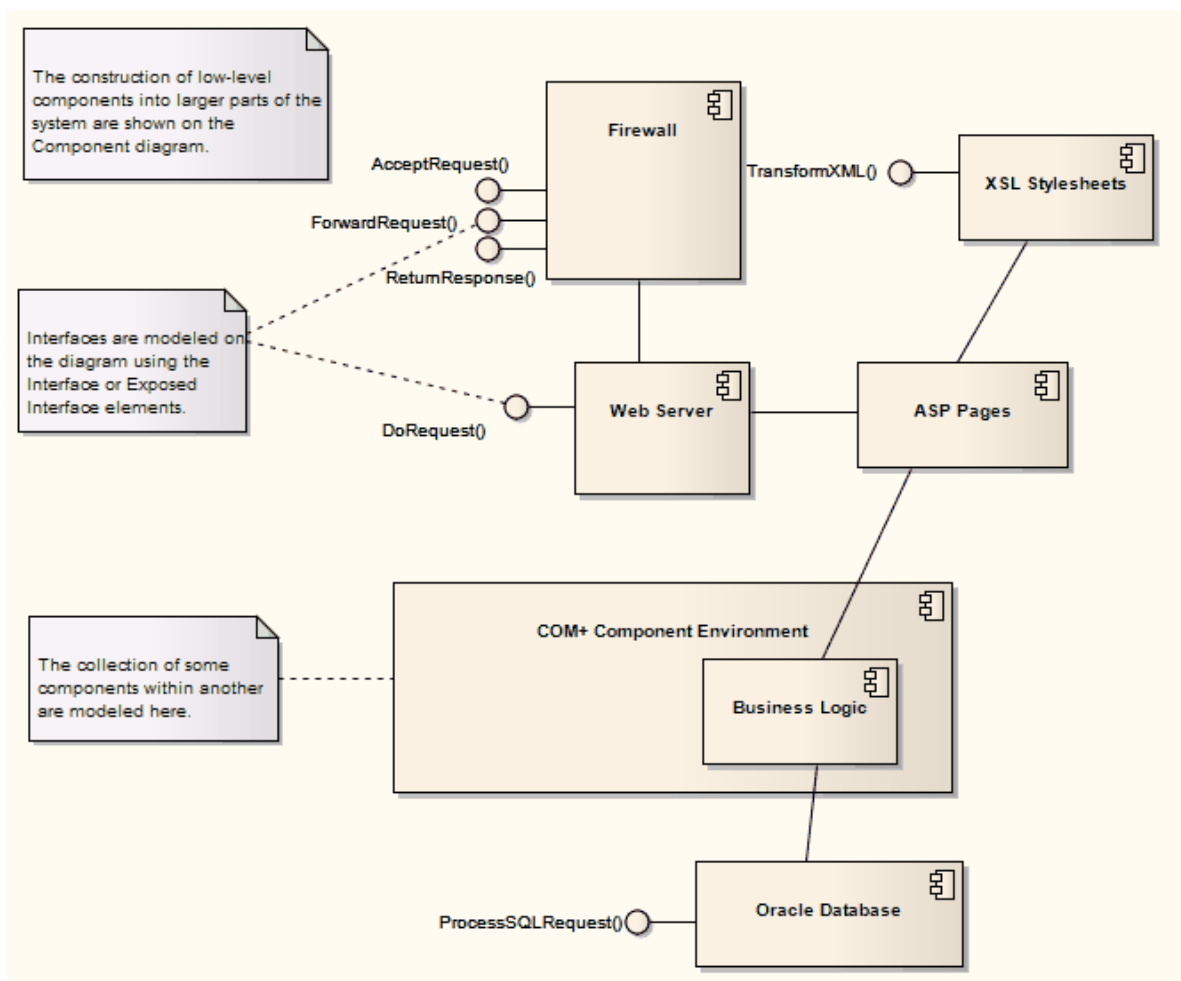
Learn more

- [Data Models](#) ¹⁶²⁵
- [UML Database Modeling](#)

5.2.2.7 Component Model Template

The *Component model* defines how Classes, Artifacts and other low level elements are collected into high level components, and describes the interfaces and connections between them.

Components are compiled software artifacts that work together to provide the required behavior within the operating constraints defined in the Requirements model.



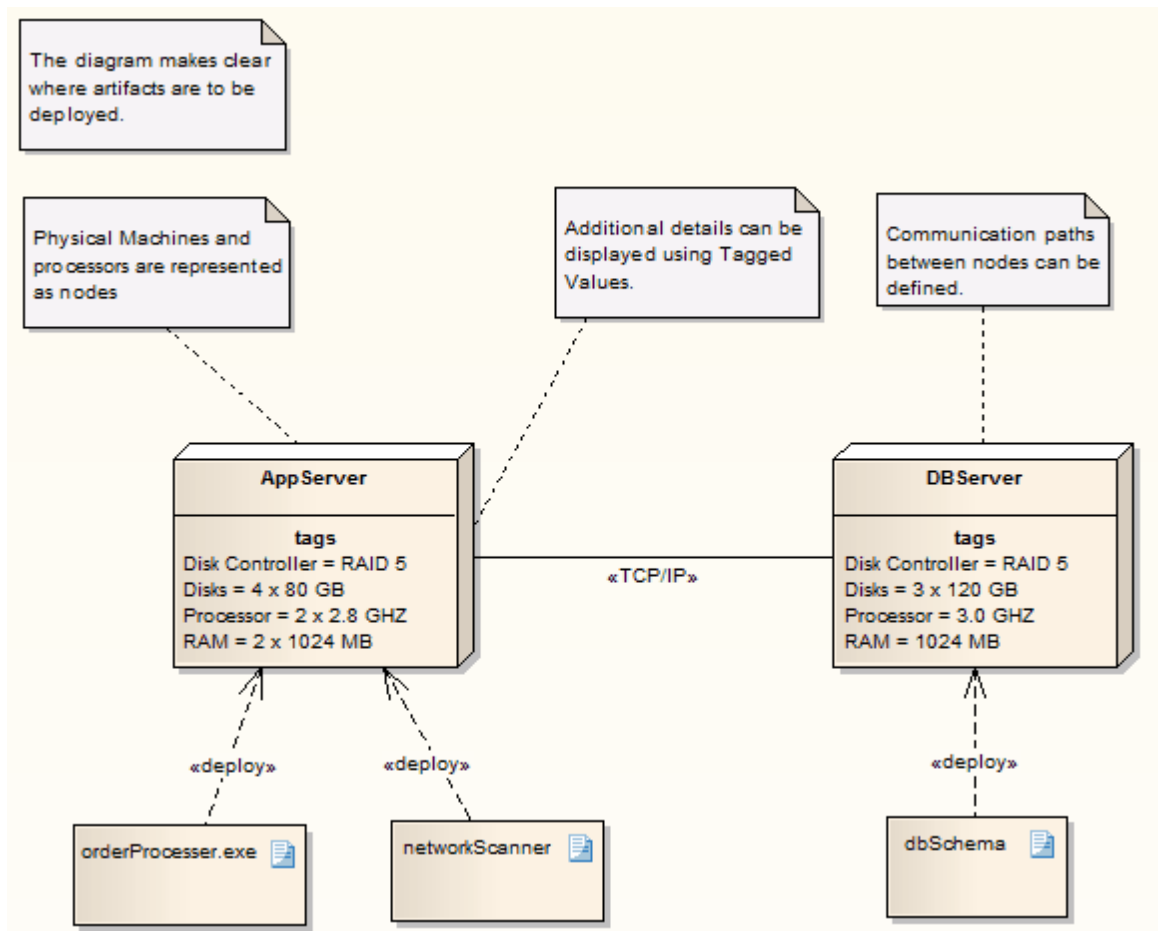
Learn more

- [Components](#) ^[1187]
- [Component Diagram](#) ^[1023]
- [The Component Model](#)

5.2.2.8 Deployment Model Template

The *Deployment model* describes how and where a system is to be deployed.

Physical machines and processors are represented by Nodes, and the internal construction can be depicted by embedding Nodes or Artifacts. As Artifacts are allocated to Nodes to model the system's deployment and roll out, the allocation is guided by the use of deployment specifications.

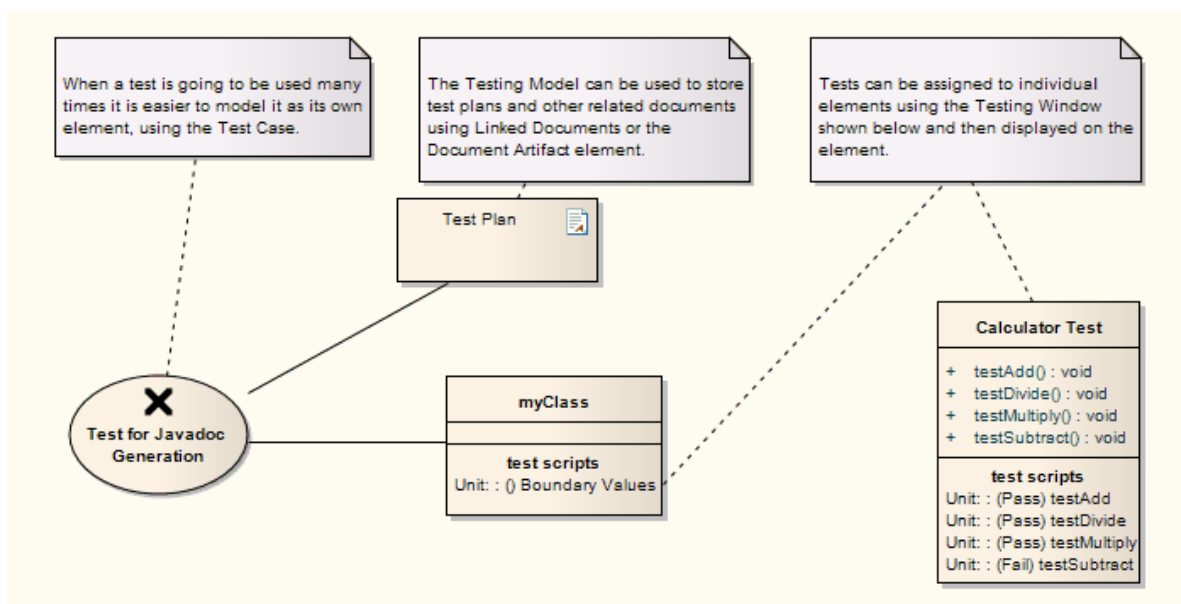


Learn more

- [Nodes](#) ^[1196]
- [Artifacts](#) ^[1179]
- [Deployment and Roll Out](#) ^[81]
- [Deployment Diagram](#) ^[1020]
- [Compartments - Tagged Values](#) ^[842]
- [The Physical Model](#)

5.2.2.9 Testing Model Template

The *Test model* describes and maintains a catalogue of tests, test plans and results that are executed against the current model.

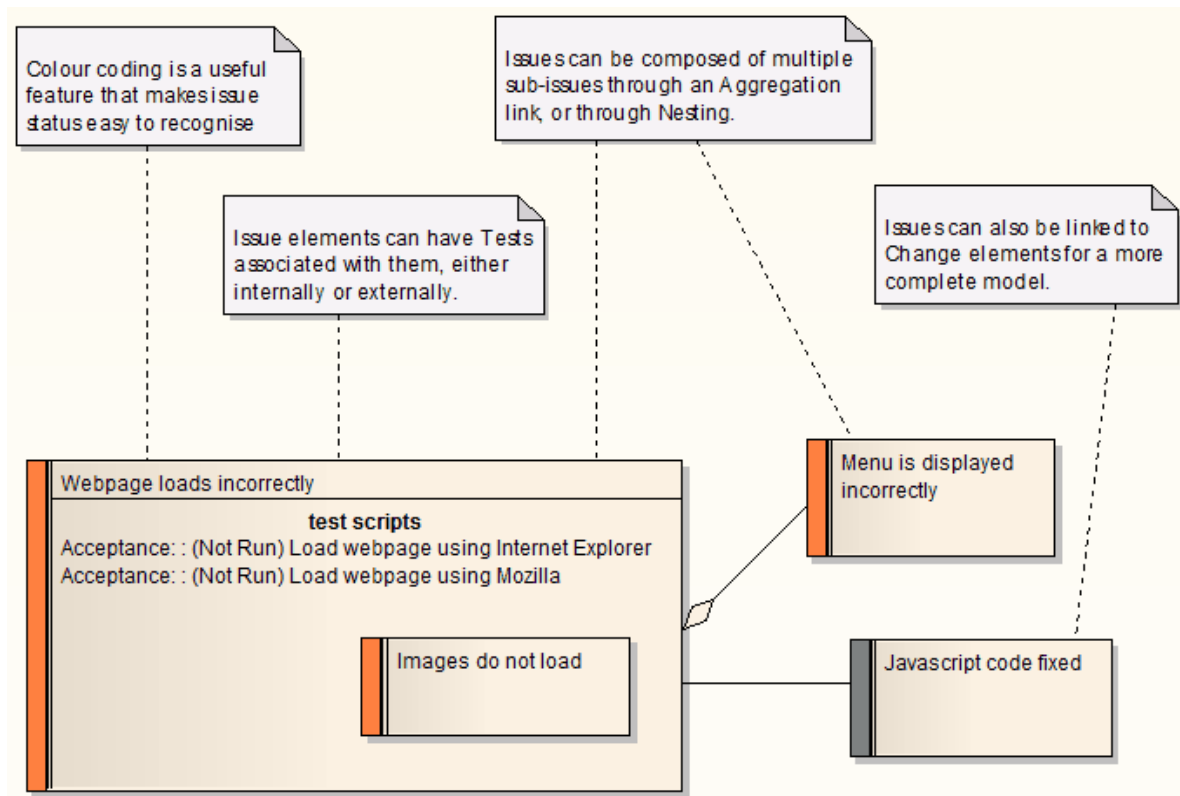


Learn more

- [Testing](#)^[221]
- [Test Case](#)^[167]
- [Show Test Script Compartments](#)^[222]
- [Testing Support in Enterprise Architect](#)

5.2.2.10 Maintenance Model Template

The *Maintenance model* enables visual representation of issues arising during and after development of a software product. The model can be enhanced with the integration of change elements and testing.



Learn more

- [Development](#)^[2236]
- [Color Code External Requirements](#)^[1477]

5.2.2.11 Project Model Template

The *Project model* details the overall project plan, phases, milestones and resourcing requirements for the current project.

Project Managers can use Enterprise Architect to assign resources to elements, measure risk and effort and to estimate project size. Change control and maintenance are also supported.

The Project Model details the overall project plan, phases, milestones and resourcing requirements for the current project.

[Read About Project Management in Enterprise Architect](#)

Release Plan

Resources

The screenshot shows a light yellow background. On the left, there is a grey box with a folded corner containing text. Below it is a blue icon of a document with a magnifying glass, followed by a blue hyperlink. On the right, there are two stacked, light brown rectangular boxes representing models. The top one is labeled 'Release Plan' and the bottom one is labeled 'Resources'.

Learn more

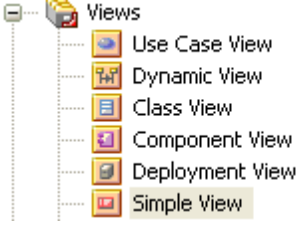
- [Project Management](#)^[44]
- [Project Manager](#) (Online Resource)

5.3 Views

The top level packages in a model (below the project root nodes) can be created as *Views*.

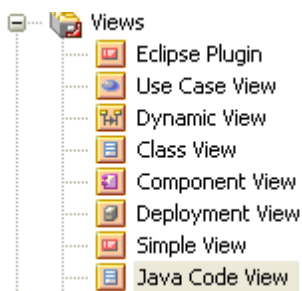
Topics

Topic	Detail	See also
Views	<p>Views are top-level packages with special package icons.</p> <p>They are used simply to subdivide the model into partitions such as Component View or Dynamic View, and are a good way to structurally extend the model depending on specific requirements and modeling techniques.</p> <p>These Views have the same concept as the Simple UML Views generated as Model Packages; however, unlike the Model Package views, basic Views:</p> <ul style="list-style-type: none"> • Do not have <i>any</i> automatically-generated components • Can be created only under a root node • Are not directly available for Requirements, Analysis or Testing, although you can develop these Views as Simple Views or within Use Case Views (see below) 	<p>Model Wizard ^[660]</p>
View Types	<p>There are six main types of View, each with their own package icon:</p> <ul style="list-style-type: none"> • Use Case View - to contain, for example, Use Case diagrams and Analysis diagrams • Dynamic View - to contain, for example, Activity diagrams, Communication diagrams, Sequence diagrams and State Machine diagrams • Class View - to contain, for example, Class diagrams, Code Engineering and Data Models • Component View - to contain, for example, Component diagrams • Deployment View - to contain, for example, Deployment diagrams • Simple View - to customize your own type of view 	<p>Use Case Diagrams ^[1030]</p> <p>Analysis Diagrams ^[1509]</p> <p>Activity Diagrams ^[1028]</p> <p>Communication Diagrams ^[1084]</p> <p>Sequence Diagrams ^[1074]</p> <p>State Machine Diagrams ^[1032]</p> <p>Class Diagrams ^[1014]</p> <p>Code Engineering ^[1733]</p> <p>Data Models ^[1625]</p> <p>Component Diagrams ^[1023]</p> <p>Deployment Diagrams ^[1026]</p>

Topic	Detail	See also
	 <p>You can use the first five categories, or devise your own based on the Simple View.</p> <p>You can create Views, rename them, move them into a different order or delete them, by right-clicking the mouse on the selected View and choosing the appropriate context menu option.</p>	Add Views ^[677] Rename Views ^[678] Delete Views ^[678]

5.3.1 Add Views

The example below shows a customized view called *Java Code View*, which has been appended to the end of the Views list.



Access **Project Browser root node context menu | Add | New View**

Reference

On the Create New View dialog, complete the fields as indicated below:

Field/Option	Action	See also
Name	Type the name of the View.	
Set View Icon Style	Click on the appropriate radio button to set the icon displayed next to the View name.	
Add to Version Control	<p>If the model root node is under version control, this checkbox displays, defaulted to selected.</p> <p>If you do not want the new View to also be under version control, deselect the checkbox.</p>	Version Control ^[312]

Field/Option	Action	See also
OK	Click on this button to add the View to the Project Browser.	

Learn more

- [Views](#) ^[676]
- [Rename Views](#) ^[678]
- [Delete Views](#) ^[678]

5.3.2 Rename Views

How to

To rename a view

Step	Action	See also
1	Right-click on the View in the Project Browser, and select the Properties context menu option. The Package Properties dialog displays.	
2	In the Name field, type the new name and click on the OK button.	

Learn more

- [Views](#) ^[676]

5.3.3 Delete Views

How to

To delete a View, if necessary

Step	Action	See also
1	In the Project Browser, right-click on the View to delete and select the Delete <viewname> context menu option. A warning that child elements are to be deleted is displayed.	
2	To delete the View and its contents, click on the Yes button. To cancel the deletion, click on the No button.	

Notes

- If you delete a View, all its contents are deleted at the same time; it cannot be restored

Learn more

- [Views](#) ⁶⁷⁶

5.4 Packages



A *package* is a container of model elements. It is represented in the Project Browser by the 'folder' icon familiar to Windows users.

This topic explores the tasks you can perform with packages, including:

- Open a package
- Add a package
- Rename a package
- Copy a package
- Drag a package onto a diagram
- Show or hide a package
- Delete a package
- List the elements in a package, with their properties, on the Package Browser

Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Update Element permission to update or delete a package

Learn more

- [Open a package](#)^[74]
- [Add a package](#)^[681]
- [Rename a package](#)^[682]
- [Copy a package](#)^[683]
- [Drag a package onto a diagram](#)^[684]
- [Show or hide a package](#)^[685]
- [Delete a package](#)^[685]
- [Package Browser](#)^[587]
- [Update Element](#)^[264]

5.4.1 Open Package in the Project Browser

How to

To open a package from the Project Browser

Step	Instruction	See also
1	Double-click on a package; the contents display in the Project Browser.	
2	Click on the + and - symbols next to the folder icon to open or close the package respectively.	

Notes

- Package contents are arranged alphabetically and elements can be dragged from one package to another using the mouse

5.4.2 Add a Package

How to

To add a package to the model hierarchy

Step	Instruction	See also
1	In the Project Browser, select the package or view under which to add a new package.	
2	Right-click on the folder icon within the Project Browser, and select the Add Add Package context menu option. The New Model Package dialog displays.	
3	In the Package Name field type the name of the new package.	
4	To immediately create a diagram for the package, leave the Automatically add new diagram checkbox selected. To avoid creating a diagram, deselect the checkbox.	
5	If you are adding a package to a parent package that is under version control, the Add to Version Control option displays, with the checkbox selected. Deselect the checkbox to exclude the new package from version control; otherwise leave it selected.	Version Control ^[312]
6	Click on the OK button. The new package is inserted into the tree at the current location and, if you left the Automatically add new diagram checkbox selected, the New Diagram dialog displays.	New Diagrams ^[724]

Step	Instruction	See also
7	If you have selected to put the package under version control, the Package Control Options dialog displays. Complete this dialog as required.	Configure Packages [426]

Notes

- You can also add a new package element by dragging the **Package** icon from the Toolbox into a diagram; in this case the package is created under the diagram's owning package, and is created with a default diagram of the same type as that in which the package is created
- In a multi-user environment, other users do not see the change until they reload their project

Learn more

- [Refresh View/Mode](#) [242]

5.4.3 Rename a Package**How to**

To rename a package

Step	Instruction	See also
1	In the Project Browser, right-click on the package to rename and click on the Package Properties context menu option.	
2	In the Name field, type the new name.	
3	Click on the OK button.	

Notes

- Alternatively, highlight the package to rename, and press **(F2)**
- In a multi-user environment, other users do not see the change until they reload their project

Learn more

- [Refresh View/Mode](#) [242]

5.4.4 Copy a Package

Enterprise Architect enables you to quickly duplicate a complete package, including its child packages, elements and diagrams.

Topics

Topic	Detail	See also
Inserting a copy of a package	<p>You can insert the copy of a package:</p> <ul style="list-style-type: none"> • Under the same parent • Under one or more other packages in the same model or project, or • Under one or more other packages in any other model or project <p>This procedure is effectively the same as exporting and importing the package XMI file, with the Strip GUIDs checkbox selected.</p> <p>You would tend to use this procedure for copying sections of a model within the project rather than reproducing an entire model or project, although copying these larger structures is equally feasible.</p>	Package XMI files ^[44]
Copying a package	<p>In the Project Browser, right-click on the required package and select the Copy Package to Clipboard context menu option (or click on the package and press (Ctrl+C)).</p> <p>The Copy Package to Clipboard dialog briefly displays until the copy operation completes.</p>	
Pasting a package	<p>In the Project Browser, right-click on the package into which to paste the copied package, and select the Paste Package from Clipboard context menu option (or click on the package and press (Ctrl+V)).</p> <p>The Paste Package from Clipboard dialog briefly displays until the paste operation completes.</p> <p>The target package is expanded to expose the pasted package in the Project Browser; if you are pasting the package within the same model as the copied source, the source parent package is also collapsed.</p> <p>If the target package already contains:</p> <ul style="list-style-type: none"> • a package with the same name as the pasted package, the pasted package name has the suffix - <i>Copy</i> • a package with the same name as the pasted package including the - <i>Copy</i> suffix, the suffix becomes - <i>Copy1</i> (or - <i>Copy2</i>, - <i>Copy3</i> and so on, as copies of the 	

Topic	Detail	See also
	<p>package accumulate in the target parent package)</p> <p>You can keep the same package name as the source, or you can rename the package either by clicking twice on it and editing the name in the Project Browser, or by double-clicking on it and editing the name in the Properties dialog.</p>	

Notes

- A copy of a package does not have the external cross references of the source package; that is, the following connectors are discarded:
 - Connectors coming from packages and elements outside the package being copied, into the package being copied
 - Connectors going to packages and elements outside the package being copied, from the package being copied
- You cannot paste a package into a parent package that is locked by another user or that is checked in; the **Paste...** option is grayed out in the context menu

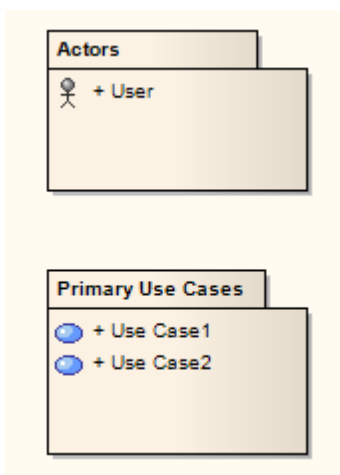
Learn more

- [Locking Packages](#) ^[273]
- [Checking In/Checking Out Packages](#) ^[353]

5.4.5 Drag a Package Onto a Diagram

You can drag a package element from the Project Browser onto the current diagram, as an icon of the package listing its contents. This is a useful feature to help organize the display and documentation of models.

The following illustration shows how a package is displayed in a diagram; note the child Actor and Use Case icons.



Learn more

- [Show or Hide Package Contents](#)^[685]

5.4.5.1 Show or Hide Package ContentsHow to

To show or hide the contents of packages in a diagram

Step	Instruction	See also
1	Load a diagram.	
2	Double-click in the background area to open the Diagram Properties dialog.	
3	Click on the Elements tab.	
4	Select or clear the Package Contents checkbox as required.	
5	Click on the OK button.	

5.4.5.1.1 Delete a PackageHow to

To delete a package

Step	Instruction	See also
1	Right-click on the package in the Project Browser, and select the Delete context menu option. A confirmation prompt displays.	
2	Click on the OK button.	

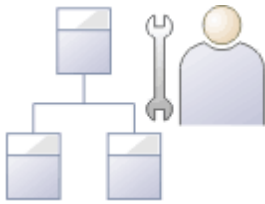
Notes

- Deleting a package also deletes all contents of the package, including sub-packages and elements; make very sure that you really want to do this before proceeding
- In a multi-user environment, other users do not see the change until they reload their project

Learn more

- [Refresh View/Model](#)^[242]

5.5 Diagrams



Diagrams are collections of project elements laid out and inter-connected as required.

Enterprise Architect supports all of the UML diagrams, as well as several custom extensions. Together with the Enterprise Architect elements and connectors, these form the basis of the model. Diagrams are stored in packages and can have a parent object (optional). Diagrams can be moved from package to package.

Notes

- If the diagram display is too small to read comfortably, click on the diagram, press and hold (**Ctrl**) and use the mouse wheel to temporarily expand or reduce the display magnification.

Learn more

- [Diagram Context Menu](#)^[68]
- [Diagram Tasks](#)^[72]

5.5.1 Diagram Context Menu

To open the diagram context menu, open the required diagram and right-click on the diagram background.

Reference

Option	Usage	Shortcut	See also
Properties	Display the Diagram Properties dialog.		Diagram Properties ^[725]
Scripts	List the scripts enabled for execution directly from the diagram (does not display if no diagram scripts exist).		Scripts ^[2375]
Swimlanes and Matrix	Add Swimlanes or a Swimlanes Matrix to the diagram.		Swimlanes ^[75] Swimlanes Matrix ^[753]
Lock Diagram	Lock a diagram to protect it from inadvertent changes. This does not apply if security is enabled in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions, in which case you lock the model elements.		Lock a Diagram ^[76] Lock Model Elements ^[27]

Option	Usage	Shortcut	See also
List View Gantt View	Display the diagram contents as a Diagram List or as a Gantt chart instead of as a diagram.		Diagram List ^[59]
New Element or Connector	Insert various elements and connectors into a diagram.		Insert Elements & Connectors ^[69]
Insert Existing Element	Locate an existing element, using the Select Element dialog, and insert it into the diagram as a linked element.		
Paste Element(s) as Link Paste Element(s) as New	Paste copied element(s) as a link or as new elements.		Paste copied element ^[73]
Paste Image from Clipboard	Paste an image held on the clipboard into the diagram.	(Ctrl+Shift+ Insert)	Paste an image ^[94]
Make all elements selectable	<p>Make all the elements on the diagram selectable.</p> <p>If an element is:</p> <ul style="list-style-type: none"> • Selectable, you can move it around the diagram and perform right-click context-menu operations • Unselectable, you cannot move it around the diagram and the only right-click operation available is to make the element selectable <p>This option has no effect on double-click operations on the element, such as displaying child diagrams.</p>		
Modify Z Order	<p>Re-order overlapping objects in the diagram.</p> <p>This option both:</p> <ul style="list-style-type: none"> • Numbers the objects in the diagram by Z-order, in red, and • Lists the objects by Z-order in the Modify Z Order dialog <p>In the dialog, you select the required object and</p>		

Option	Usage	Shortcut	See also
	<p>move it up or down in the Z-order using the Up and Down arrow keys, each click on an arrow key changes the Z-order by 1.</p> <p>When you select the object in the list, that object is also selected on the diagram and its Z-order number is shown in blue.</p> <p>Click on the OK button to save your changes.</p>		
Save as Profile	Save the current diagram as a Profile.		Export a Profile ^[1306]
Import from source file(s)	Import, or reverse engineer, source code (not available in the Desktop edition).		Import, or reverse engineer, source code ^[1792]
Import DB schema from ODBC	Import database tables from an ODBC data source (not available in the Desktop edition).		Import database tables from an ODBC data source ^[1973]
Save Current Changes	Save any changes to the current diagram.	(Ctrl+S)	
Execute Simulation	<p>Provides three options for performing a model simulation on a behavioral diagram:</p> <ul style="list-style-type: none"> • Interpreted - Perform dynamic execution of a simulation (Corporate and Extended Editions) • Manual - Step through a simulation manually (the only option available in the Professional edition) • Using Script - If simulation scripts exist for the diagram, lists the scripts available; if no scripts exist, the option does not display 		Model Simulation ^[2100] How It Works ^[2102] Set Up Simulation Script ^[2109]
Help	View the Enterprise Architect Help on the type of diagram currently displayed.		

Notes

- Not all the menu options described appear on all diagram context menus

5.5.1.1 Insert Elements and Connectors

When you click on the **New Element or Connector** option on the **Diagram** context menu, a list of elements and connectors displays. This is the **Toolbox Shortcut** menu.

Topics

Topic	Detail	See also
Structure	<p>The structure of this list is as follows:</p> <ul style="list-style-type: none"> • Other - expands to offer options to select elements and connectors from diagram types other than either the current diagram type or pinned Toolbox pages • The expanded list of elements and connectors for the current diagram type • Collapsed lists of elements and connectors for pages that have been pinned in the Toolbox; if an MDG Technology: <ul style="list-style-type: none"> • is active • automatically pins Toolbox pages, and • has pages that redefine UML or Extended pages <p>The MDG Technology pages override the UML or Extended pages, which are not shown</p> • (At the end) Common - expands to display a list of the common elements and connectors 	

Notes

- The list of elements and connectors displays instead of the Diagram context menu, if you press **(Ctrl)** or **(Shift)** as you right-click the mouse
- You can also display the list by pressing **(Spacebar)** or **(Insert)** while the cursor is on the diagram

Learn more

- [Toolbox Shortcut Menu](#) 

5.5.2 Print Preview


This topic describes how to display a preview of an Enterprise Architect diagram or document prior to printing it.

Access **File | Print Preview**

Use to

- Preview pages before printing them


Notes

- When you select the **File | Print Preview** menu option, the display initially shows the first two pages on one screen, with no scroll bar
- To toggle between the two-page display and a single-page display, click on the  icon in the preview screen toolbar; in either mode, you can use the forward and back arrows to scroll through the pages of the diagram
- To display more than two pages on one screen, up to a maximum of ten pages, click on the **Zoom Out** button in the Preview screen toolbar; the screen now includes the vertical scroll bar, which you can also use to scroll through the pages of the diagram

5.5.3 Diagram View

The Diagram View is the main workspace window that enables you to create and display diagrams.

Topics

Topic	Detail	See also
Opening a Diagram	<p>To open a diagram, double-click on the diagram name in the Project Browser; you can open further diagrams in the same way.</p> <p>You can also open a diagram from within an already-open diagram by clicking on either hyperlinks or elements that contain other diagrams.</p> <p>You can open many diagrams, but you can view only one at a time.</p>	
Caption Bar	<p>Across the top of a diagram is the diagram caption bar, which includes the following:</p> <ul style="list-style-type: none"> • Icon and text label for the diagram type • Diagram name • Date and time the diagram was created • Date and time the diagram was last modified • The current magnification (zoom) of the diagram • The diagram page size, in pixels • A toggle icon to display or hide the Format Toolbar (see below) • A drop-down arrow that lists the currently-open diagrams; click on a diagram name to switch to that diagram, or to redisplay the hidden Start Page • The Window 'close cross'; click on this to close the displayed diagram 	
Format Toolbar	<p>Underneath the Caption Bar is the Format Toolbar, which enables you to apply a range of functions to the format and appearance of the elements in your diagram.</p>	Format Toolbar 

Topic	Detail	See also
	You can hide or reveal this toolbar by clicking on the 'double chevron' icon in the Caption Bar (third icon from the right).	
Diagram View	<p>Use the Diagram View to build model relationships and elements.</p> <p>Within the diagram, you can create new elements, drag in existing elements and generally organize the elements and relationships.</p> <p>Most work on elements is carried out in the Diagram View, so understanding how it works and how to manipulate elements is essential; use the example project supplied with Enterprise Architect to explore the capabilities and behavior of the Diagram View.</p>	

Notes

- You can also use the Diagram List to manipulate elements

Learn more

- [Diagram List](#)^[596]
- [Diagram Tabs](#)^[695]

5.5.3.1 Format Toolbar

The Format Toolbar can change the appearance of one or more selected elements by altering the font, color, line width and fill color. This can help to group, differentiate or make an element stand out.

You can hide or reveal this toolbar by clicking on the 'double chevron' icon (third icon from the right) in the Caption Bar at the top of the Diagram View.

An abbreviated Format Toolbar can also be displayed temporarily beside an element or connector. For elements, select one or more elements on the diagram, then click on the paintbrush icon beside the element. For connectors, right-click on the connector and the Format Toolbar appears above the context menu.

Use to

- Modify the appearance of elements and connectors on a diagram
- Save a set of modifications as a user defined style
- Re-apply styles to other elements and connectors

Reference



Action	Usage	Shortcut	See also

Set Font	Set the text font, style, size and effects for the selected element(s) using the Font dialog.		Set Element Font ⁸³⁹
Text Color	Set the text color for the selected element(s). This option overrides the color setting of the Set Font option.		
Fill Color	Set the fill color of the selected element(s).		
Line Color	Set the line color of the selected connector or element border.		
Line Width	Set the line width of the selected connector or element border; choose a value between 1 (thinnest) and 5 (thickest).		
Apply Style	Apply a copied style to the selected elements or connector. The style is copied using the Get Style option; if this is not used, the style defaults to the current setting of the Style List option.		
Get Style	Copy the current style of the selected element or connector.		
Style List	Apply a style from the saved style list to the selected element or connector. These styles are saved to the list using the Save Style option.		
Save Style	Click on the required element, click on the down arrow and select the appropriate option: <ul style="list-style-type: none"> • Save the style of the element to the Style List as a new style; enter the name of the style when prompted to do so • Save the (updated) style of the element under its current style name in the Style List • Refresh the Style List with any changes to the existing styles • Delete the style of the selected element from the Style List (refresh the list after selecting this option) 		

Align Left	Align the selected elements on the left border of the last-selected element.	Ctrl + Alt + Left Arrow	
Align Right	Align the selected elements on the right border border of the last-selected element.	Ctrl + Alt + Right Arrow	
Align Top	Align the selected elements on the top edge of the last-selected element.	Ctrl + Alt + Up Arrow	
Align Bottom	Align the selected elements on the bottom edge of the last-selected element.	Ctrl + Alt + Down Arrow	
Move to Top	Bring the selected element(s) to the top of the Z order.		Diagram Context Menu [687]
Send To Back	Move the selected element(s) to the bottom of the Z order.		
Diagram Auto-layout	Auto-layout the diagram (not for Behavioral diagrams) in one of the standard layouts, or modify a standard layout and apply that.		Layout Diagrams [769]
Diagram Properties	Display the Diagram Properties dialog for the diagram.	F5	Diagram Properties [725]
Delete	Delete the selected element(s).	Ctrl + D	
Add or Remove Buttons	<p>This drop-down arrow at the right end of the toolbar enables you to:</p> <ul style="list-style-type: none"> • Display or hide icons currently defined for the toolbar, or • Customize the toolbar to include other icons, to initiate commands found on other toolbars or totally new commands <p>(As an example, the last nine options on the Format toolbar are also found on the Diagram toolbar, which contains other options that could be duplicated here).</p>		The Customize Dialog [153] Diagram Toolbar [145]

Notes

- To set the global appearance of all elements throughout a model, use the Options dialog; select the **Tools | Options** menu option, then select **Standard Colors** and **Diagram | Appearance** from the

options tree

- To override the global appearance and define a default appearance of a selected element (or several selected elements) on all diagrams on which it occurs, set the Default Appearance for that element; right-click the element and choose the **Appearance | Default Appearance** context menu option


Learn more

- [Standard Colors](#) ^[546]
- [Appearance](#) ^[550] (Diagram)
- [Set An Element's Default Appearance](#) ^[817]

5.5.4 Diagram Tabs

Diagram tabs are located, by default, at the bottom of the Diagram View. You can also move them to the top of the Diagram View, above the Status Bar.

Topics

Topic	Detail	See also
Access	<p>Each time you open a diagram, the diagram name and diagram type symbol are shown in the tab for easy identification and access.</p>  <p>You can manipulate the tabs using the Diagram Tabs context menu (below) or the View Tab Switcher window - press (Ctrl+Tab).</p>	View Tab Switcher ^[697]
The Diagram Tabs Menu	To access the Diagram Tabs context menu, right-click on an appropriate tab.	

Reference

Option	Usage	Shortcut	See also
Save Changes to '<tab name>'	Save the changes made to the diagram.		
Save All	Save the model.		
Reload '<tab name>'	<p>Reopen the diagram without the unsaved changes; that is, revert to the state before any changes were made.</p> <p>Refresh the diagram from the repository, to show any changes made by other users in a shared model.</p>		Refresh the diagram ^[242]

Option	Usage	Shortcut	See also
Close '<tab name>'	Close the diagram; Enterprise Architect prompts you to save changes to the diagram. You can also close the tab by clicking the <i>middle</i> mouse button/scroller.		
Close All	Close all open diagrams; Enterprise Architect prompts you to save any diagrams with unsaved changes.		
Close All Except '<tab name>'	Close all diagrams except for '<tab name>'; Enterprise Architect prompts you to save any diagrams with unsaved changes.		
Close All <view type>	(Where several views of the same type can be opened at the same time, such as diagrams, RTF documents, or text editors.) Close all views of the same type as the selected tab, leaving views of other types still open.		
Hide	(Start Page only) Hide the Start Page. To redisplay the Start Page, use the drop-down menu at the top right of the Diagram View, or select the Window Show Start Page menu option.		

Notes

- The Component View tab is white with bold text; this means that the Component View diagram is the diagram currently displayed in the Diagram View
- The Use Case View tab has an asterisk; this means that there are unsaved changes in the Use Case View diagram
- There are also tabs for other types of view, such as the Start Page (as shown), RTF reports, code editors, the Model Search and the Relationship Matrix
- You can change the sequence of tabs by clicking on each tab and dragging it to a different position, as required; however, you cannot change the position of the Start Page tab - when displayed, it always sits at the left-hand end of the sequence

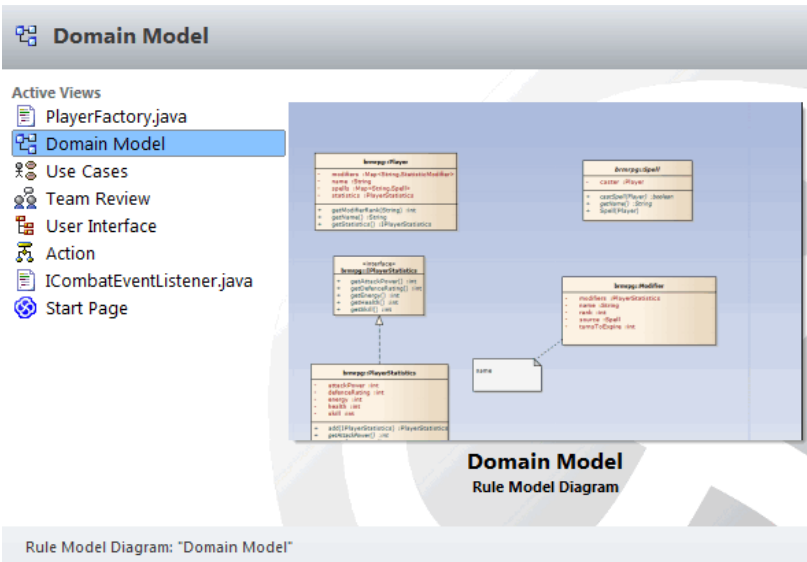
Learn more

- [General Settings](#) ^[544]

5.5.4.1 View Tab Switcher

The **View Tab Switcher** facilitates navigation between tabs of the Diagram View (such as Diagrams, Code Editors and the Model Search screen) that are currently open in Enterprise Architect.

Topics

Topic	Detail	See also
<p>Usage</p>	 <p>The screenshot shows the 'Domain Model' tab selected in the View Tab Switcher. The diagram area displays a UML class diagram with the following classes and relationships:</p> <pre> classDiagram class Player { +modifiers :Map<String,StatisticsModifier> +name :String +skills :Map<String,Spell> +statistics :PlayerStatistics +getModifiers():Map<String,StatisticsModifier> +getSkills():Map<String,Spell> +getStatistics():PlayerStatistics } class PlayerStatistics { +modifiers :PlayerStatistics +name :String +skills :String +getModifiers():PlayerStatistics +getName():String +getSkills():String } class PlayerStatistics2 { +modifiers :PlayerStatistics +name :String +skills :String +getModifiers():PlayerStatistics +getName():String +getSkills():String } class Modifier { +modifiers :PlayerStatistics +name :String +spell :Spell +nameOfDamage :int } PlayerStatistics < -- PlayerStatistics2 PlayerStatistics2 ..> PlayerStatistics : same </pre> <p>Rule Model Diagram: "Domain Model"</p>	
<p>Activating a tab</p>	<p>To display the View Tab Switcher, press (Ctrl + Tab).</p> <p>The View Tab Switcher lists the currently opened tabs, in order of most recently activated to least recently activated; the selection cursor is on the tab that was activated just before the currently active one.</p> <p>Details of the selected tab are displayed on the right hand side of the screen, and include:</p> <ul style="list-style-type: none"> • A preview thumbnail of the tab, and • A description of the tab and its contents <p>Continue holding the (Ctrl) key to keep the window open; to move the selection cursor:</p> <ul style="list-style-type: none"> • Down the list of tabs, press the (Tab) key • Up the list, press the (Shift+Tab) keys <p>Alternatively, use the arrow keys.</p>	

Topic	Detail	See also
	required tab. Activating a tab moves it to the top of the Active Views list, whereas closing a tab removes it from the list completely.	

5.5.5 Diagram Toolbox

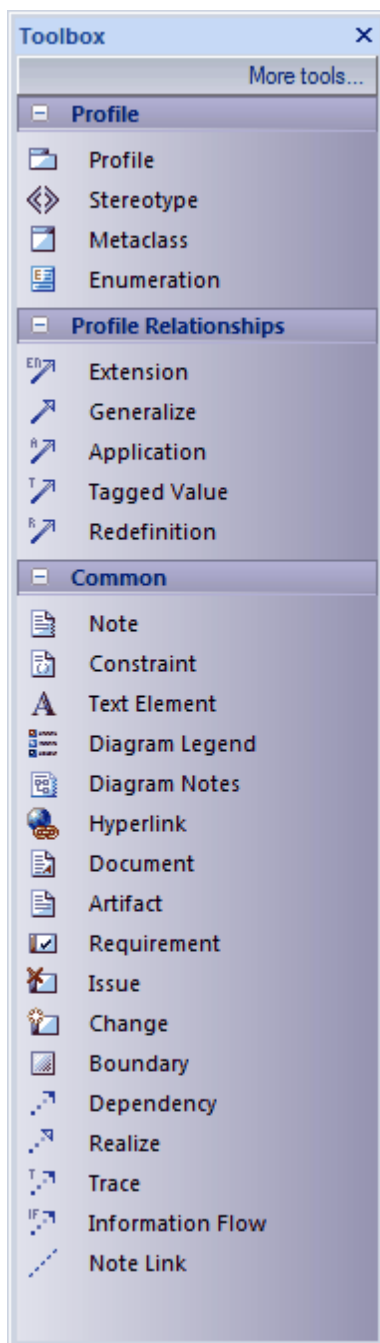
The Enterprise Architect Toolbox is a panel of icons that you use to create elements and connectors on a diagram.

Within the Toolbox, related elements and connectors are organized into pages, each page containing the elements or connectors used for a particular type of diagram. The diagrams include standard UML diagrams, Enterprise Architect Extended diagrams, and any MDG Technologies and Profiles that you have added to Enterprise Architect.

When you open a diagram, the Toolbox automatically provides the element and relationship pages corresponding to the diagram type. This does not prevent you using elements and connectors from other pages in a given diagram, though some combinations might not represent valid UML.

Access **View | Toolbox (Alt+5)**

Example



Topics

Topic	Detail	See also
Display specific pages	<p>To display specific pages in the Toolbox, select the More Tools option at the top of the Toolbox and select the appropriate UML, Extended or customized diagram type from the menu.</p> <p>In most cases, three pages display:</p> <ul style="list-style-type: none"> • <type> elements • <type> Relationships and 	

Topic	Detail	See also
	<ul style="list-style-type: none"> • Common <p>If you select the <default> option, you display only the Common page.</p>	
Customize the Toolbox	You can customize the Toolbox pages by pinning them within the Toolbox, or by adding MDGTechnologies and UML Profiles to the Toolbox	Toolbox Appearance Options ^[701] MDG Technologies ^[1285] UML Profiles ^[1277]

How to

To create elements and connectors from the Toolbox

Step	Instruction	See also
1	In the Project Browser, double-click on the icon against the required diagram. The diagram opens with the appropriate Toolbox pages for that diagram type. If you want a different set of elements and connectors, click on More tools and select the appropriate diagram type as explained above.	
2	Click on the required item; for example, the Class element or Associate relationship.	
3	For element items, click anywhere on the diagram to place the new element.	
4	For connector items, drag the cursor between the source and target elements on the diagram; the solid border of the elements changes to a hatched border as you pass the cursor over them, indicating the source and potential target elements. To add bends to the connector, press (Shift) as you change the drag direction of the cursor. Alternatively, drag from the source element to an empty area of the diagram; the Quick-linker enables you to create the target element.	Quick-linker ^[792]
5	Edit the element properties or connector properties, as required.	Properties Dialog ^[843] Connector Properties ^[961]

Notes

- Dropping a Package element from the Toolbox into a diagram creates a new package in the Project Browser, and a default diagram of the same type as the current diagram
- If you are creating several elements of one type, after creating the first just press (**Shift+F3**) or (**Ctrl**) + **click** to create the next element of that type; for connectors, click on the source element and press (**F3**) to create another connector of the same type
- You can change an unsteretyped element to one of its stereotyped elements by dragging the stereotyped element from the Toolbox onto the unsteretyped element in the diagram; for example, you can stereotype a Class by dragging a Table element or a Profiled Class element onto it
- As you drag the stereotyped element, you can press (**Ctrl**) to apply the element stereotype to the unsteretyped element, or (**Shift**) to add the stereotyped element as a new element; otherwise a small context menu displays offering these actions as options
- If the diagram element already has the stereotype, you can also drag the Toolbox element onto it to synchronize the element's stereotype Tagged Values
- The Toolbox can be docked on either side of the diagram, or free floated on top of the diagram to expose more surface for editing
- You can also hide and show the whole Toolbox using the  (**Show Element Toolbox**) button on the Workspace Views toolbar
- Enterprise Architect provides Toolbox pages for a wide range of MDG Technologies, such as ArchiMate, BPEL, BPMN, Data Flow Diagrams, ICONIX and Mind Mapping, as part of the initial install

Learn more

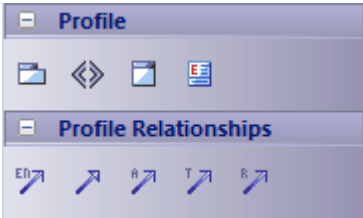
- [Stereotype Dialog](#) ^[1264]
- [Synchronize Tags and Constraints](#) ^[1281]
- [ArchiMate](#) ^[1621]
- [BPEL](#) ^[1570]
- [BPMN](#) ^[1545]
- [Data Flow Diagrams](#) ^[1506]
- [ICONIX](#) ^[1925]
- [Mind Mapping](#) ^[1503]

5.5.5.1 Toolbox Appearance Options

You can modify the appearance of the Toolbox pages in several ways, through the context menu.

Access **Right-click on the Toolbox page**

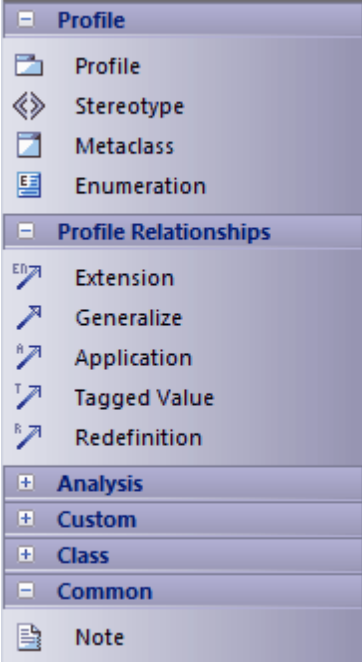
Topics

Topic	Detail	See also
Hide or Show element or relationship labels	<p>Select the Hide Labels (or Show Labels) context menu option.</p> <p>The icons in the page then 'wrap' within the page, without text labels.</p>  <p>When you hide the labels, you can display the label of an individual element or relationship by moving the cursor over the icon.</p>	
Pinning pages to display All the time	<p>Select the Pin in Toolbox context menu option (not available on the Common page, which displays all the time anyway).</p> <p>For example, if you 'pinned' the Class elements page, and switched to the Communication pages, the Toolbox would include a collapsed Class elements page below the Communication pages.</p>	
Unpinning pages	<p>To unpin the page so that it displays only in its own Toolbox group, select the Unpin from Toolbox context menu option.</p>	
Collapsing pages	<p>To collapse a page to just show the heading (<type> elements, <type> Relationships or Common), click on the 'minus' box at the left of the page heading; alternatively, right-click on the page and select the Collapse context menu option.</p> <p>To expand the page again, click on the heading.</p>	

How to

To tailor the Toolbox to list all pages you require at the same time

Step	Instruction	See also
1	<p>Click on the Set Toolbox Visibility context menu option.</p> <p>The Visible Toolbox Pages dialog displays.</p>	
2	<p>By default, the dialog lists the element pages only, in the order: UML pages, Extended pages, MDG Technology pages.</p> <p>To include the corresponding relationship pages, select the Show Relationship Pages checkbox at the bottom of the dialog.</p>	

Step	Instruction	See also
3	<p>For each page to display on the Toolbox, select the Visible checkbox.</p> <p>Deselect the checkbox if you no longer require a page to be displayed.</p>	
4	<p>When you have defined the list of pages to display, click on the OK button.</p> <p>The pages you have selected are pinned to the Toolbox in a collapsed state, underneath the current diagram-type pages.</p> 	
5	<p>To expand a page, click on the heading.</p> <p>To remove an individual page, expand it, right-click on it, and select the Unpin from Toolbox context menu option.</p>	

Notes

- On a Toolbox page for an MDG Technology or UML Profile, if you right-click directly on a profile element an additional option - **Synchronize Stereotype** - is available at the end of the context menu; this enables you to synchronize the Tagged Values and constraints for all elements created from the selected profile element
- MDG Technologies can impose their own Toolbox page visibility; for example, if ICONIX is the active technology, all six ICONIX pages are automatically exposed in the Toolbox
- If the active Technology pages duplicate UML or Extended pages (as the ICONIX pages do), the pinned Technology pages override and replace the pinned UML and Extended pages; for example, if ICONIX is active and you have pinned the Extended Analysis page, the Analysis page in the list is the *ICONIX-*

defined page, not the Extended Analysis page

Learn more

- [Synchronize Tags and Constraints](#) 

5.5.5.2 Toolbox Shortcut Menu

To add elements and connectors into a diagram, you can access the **Toolbox** shortcut menu instead of employing the full graphical Toolbox.

Topics

Topic	Detail	See also
Usage	<p>The menu provides options to select:</p> <ul style="list-style-type: none"> • Elements specific to the current diagram type • Relationships specific to the current diagram type • Elements and relationships from any pages pinned in the Toolbox • Common elements and relationships • Elements and connectors for other diagram types <p>The advantage of using the Toolbox shortcut menu is that it provides an increased amount of the workspace to be used for diagramming rather than to display fixed (instead of pop-up) menus.</p>	

How to

To use the Toolbox shortcut menu

Step	Action	See also
1	Open a diagram.	
2	<p>Either:</p> <ul style="list-style-type: none"> • Click on the diagram background and press (Insert) or (Spacebar) • Press and hold (Ctrl) or (Shift) and right-click on the diagram background • Right-click on the diagram background and select the New Element or Connector context menu option <p>The shortcut menu displays, listing the current diagram-type elements and connectors.</p>	
3	If necessary, select the Other option or a pinned Toolbox page option to list elements	

Step	Action	See also
	and connectors for a different diagram type.	
4	Select the element or connector to include in the diagram; the object is added to the diagram. If you select the Other context menu option, the final menu in the sequence offers the Activate <Type> Toolbox option; this opens and activates the corresponding page in the Toolbox, if the Toolbox is visible.	

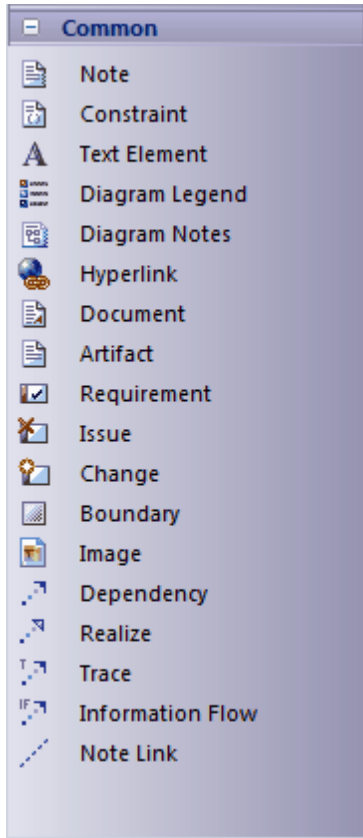
Notes

- As with the Toolbox itself, if an MDG Technology:
 - Is active
 - Automatically pins Toolbox pages, and
 - Duplicates UML or Extended pages

The pinned Technology pages override and replace the pinned UML or Extended pages in the initial **Toolbox** shortcut menu













5.5.5.3 Common Group







The Common page of elements and relationships is displayed at the bottom of every other set of pages. It contains the elements and relationships that can be used on any diagram.



Toolbox Elements and Connectors:

Click on the elements and connectors below for information.

Common Elements	Common Connectors
 Note	 Dependency
 Constraint	 Realize
 Text Element	 Trace
 Diagram Legend	 Information Flow
 Diagram Notes	 Note Link
 Hyperlink	
 Document	

Common Elements	Common Connectors
 Artifact	
 Requirement	
 Issue	
 Change	
 Boundary	
 Image	

5.5.5.4 Use Case Group

Use Case elements are used to build Use Case models. These describe the functionality of the system to be built, the requirements, the constraints and how the user interacts with the system.

Often, Sequence diagrams are associated with Use Cases to capture work flow and system behavior.

Topics

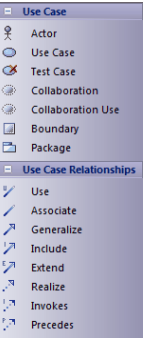
Image	Topic	Detail	See also
	Usage	<p>The Use Case group is used to model the system functionality from the perspective of a system user; the user is called an Actor and is drawn as a stick figure, although the user could be another computer system or similar.</p> <p>A Use Case is a discrete piece of functionality the system provides that enables the user to perform some piece of work or something of value using the system.</p> <p>Examples of Use Cases are: login, open account, transfer funds, check balance and logout; each of these implies some purposeful and discrete functionality the system is to provide to a user.</p> <p>A Test Case describes what must be set up in order to test a particular feature.</p> <p>The connectors available include:</p> <ul style="list-style-type: none"> • Associate (an Actor uses a Use Case) • Extend (one Use Case can extend another) • Include (one Use Case can include another) and • Realize (this Use Case might realize some business requirement) <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram; set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on</p>	Test Case <small>1677</small>

Image	Topic	Detail	See also
		the start element in the diagram and drag to the end element.	

Notes

- Invokes and Precedes relationships are defined by the Open Modeling Language (OML), and are stereotyped Dependency relationships; *Invokes* indicates that Use Case A, at some point, causes Use Case B to happen, whilst *Precedes* indicates that Use Case C must complete before Use Case D can begin

Learn more

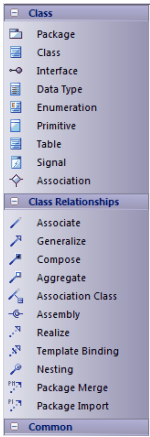
- [Use Case Models](#) ^[1030]

5.5.5.5 Class Group

The Class group can be used for Package diagrams, Class diagrams and Object diagrams: those that usually display elements concerned with the logical structure of the system, such as Objects, Classes and Interfaces.

Logical models can include domain models (high level business driven object model) through to strict development Class models (define inheritance, attributes, operations).

Topics

Image	Topic	Detail	See also
	Usage	<p>The Class group is used for creating Class models and Data models.</p> <p>Class modeling uses the Class and Interface elements, with occasional use of the Object element to model Class instances; you can add Association or Aggregation relationships.</p> <p>Use the Table element to insert a stereotyped Class for use in data modeling.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram; set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, click on the start element in the diagram and drag to the end element.</p>	Data Modeling ^[1625]

Learn more

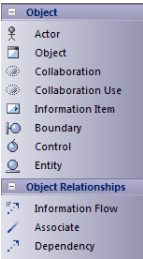
- [Package Diagrams](#) ^[1011]
- [Class Diagrams](#) ^[1014]
- [Object Diagrams](#) ^[1015]

5.5.5.6 Object Group

The Object group is used to create Object diagrams.

Object diagrams reflect multiplicity and the roles instantiated Classes could serve. They are useful in creating different cases in which relationships and Classes are applied.

Topics

Image	Topic	Detail	See also
	Usage	<p>The user is called an Actor and is drawn as a stick figure, although the user could be another computer system or similar.</p> <p>An Object is an instance of a Class.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram; set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	<p>Actor^[1107]</p> <p>Object^[1196]</p>

Learn more

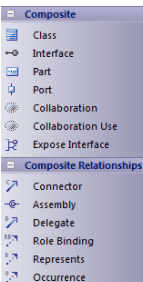
- [Object Diagrams](#)^[1015]

5.5.5.7 Composite Group

The Composite group is used for Composite Structure diagrams.

These reflect the internal collaboration of Classes, Interfaces or Components to describe a functionality or to express run-time architectures, usage patterns and the participating elements' relationships, which static diagrams might not show.

Topics

Image	Topic	Detail	See also
	Usage	<p>To add an element to the current diagram, click on the required icon, and drag it into position on the diagram; set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	

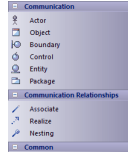
Learn more

- [Composite Structure Diagrams](#)^[1017]

5.5.5.8 Communication Group

The Communication group is used to model dynamic interactions between elements at run-time.

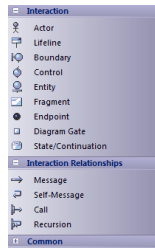
Topics

Image	Topic	Detail	See also
	Usage	<p>Communication diagrams are used to model work flow and sequential passing of messages between elements in real time; they are often placed beneath Use Case elements to further expand on Use Case behavior over time.</p> <p>The Actor element models a user of the system, while the other elements model things within the system, including standard elements (Object), user interface component (Boundary), Controller and Entity.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram; set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	Communication Diagrams ^[1084]

5.5.5.9 Interaction Group

The Interaction group is used for Interaction diagrams (Sequence, Timing, Communication or Interaction Overview), which are used to model work flow and sequential passing of messages between elements in real time.

Topics

Image	Topic	Detail	See also
	Usage	<p>The Interaction group is used to model dynamic interactions between elements at run-time; Interaction diagrams are often placed beneath Use Case elements to further expand on Use Case behavior over time.</p> <p>The Actor element models a user of the system, while the other elements model things within the system, including standard elements (Lifeline), user interface component (Boundary), Controller and Entity.</p> <p>The Message (sequence) relationship is used to model the flow of information and processing between elements.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram; set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	Sequence diagram ^[1074]

Notes

- Messages can be simple or recursive calls

Learn more

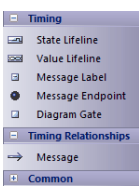
- [Sequence](#)^[1074]
- [Timing](#)^[1051]
- [Communication](#)^[1084]
- [Interaction Overview](#)^[1087]

5.5.5.10 Timing Group

The Timing group is used solely for Timing diagrams, which use a time-scale to define the behavior of objects. The time-scale visualizes how the objects change state and interact over time.

Timing diagrams can be used for defining hardware-driven or embedded software components, and time-driven business processes.

Topics

Image	Topic	Detail	See also
	Usage	<p>A Lifeline is the path an object takes across a measure of time, indicated by the x-axis.</p> <p>A State Lifeline follows discrete transitions between states, which are defined along the y-axis of the timeline; any transition has optional attributes of timing constraints, duration constraints and observations.</p> <p>A Value Lifeline shows the lifeline's state across the diagram, with parallel lines indicating a steady state; a cross between the lines indicates a transition or change in state.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram; set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	<p>State Lifeline^[1158]</p> <p>Value Lifeline^[1177]</p>

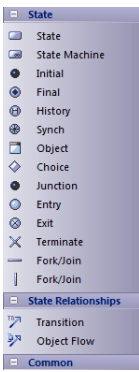
Learn more

- [Timing Diagrams](#)^[1051]

5.5.5.11 State Group

The State group is used by State Machine diagrams to show the enableable states a Class or element might be in and the transitions from one state to another. These diagrams are often placed under a Class element in the Project Browser to illustrate how a particular element changes over time.

Topics

Image	Topic	Detail	See also
	Usage	<p>The State group provides elements common to State Machine diagrams; basically the State, start and end nodes and the Object Flow relation.</p> <p>State Machine diagrams are used to model the states or conditions that elements might be in at runtime, such as active, inactive, idle, accelerating or braking.</p> <p>States can have substates; for example, accelerate or brake might be substates of active.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram; set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	

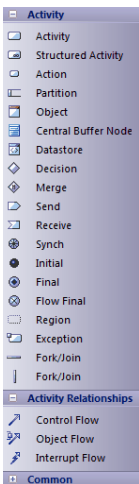
Learn more

- [State Machine Diagrams](#) 

5.5.5.12 Activity Group

The Activity group is used to model system dynamics from a number of viewpoints in Activity diagrams and Interaction Overview diagrams.

Topics

Image	Topic	Detail	See also
	Usage	<p>Activity elements enable you to describe the dynamics of the system from the point of view of activities and flows between them.</p> <p>Activities can be stereotyped as a process to display a business process icon; an Activity is some work that is carried out, which might overlap several Use Cases or form only a part of one Use Case.</p> <p>Send and Receive events are included as triggers.</p> <p>A Decision element marks a point where processing might split based on some outcome or value.</p> <p>The Flow relation models an active transition and synch points are used to split and rejoin periods of parallel processing.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram; set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	

Learn more

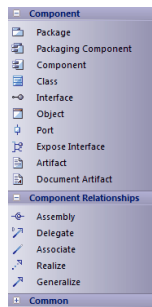
- [Activity Diagrams](#) 

- [Interaction Overview Diagrams](#) ^[1087]

5.5.5.13 Component Group

The Component group enables you to model the physical components of your system in a Component diagram.

Topics

Image	Topic	Detail	See also
	Usage	<p>The Component group contains elements related to the actual building of the system: the components that make up the system (such as ActiveX DLL's or Java beans), the Interfaces they expose, and the dependencies between those elements.</p> <p>A component is a piece of hardware or software that makes up the system; for example, a DLL or a Web Server are components that might be deployed on a Windows 2000 Server (Node).</p> <p>See the <i>Deployment Diagram</i> topic for an example.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram; set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon then click on the start element in the diagram and drag to the end element.</p>	Deployment Diagram ^[1020]

Learn more

- [Component Diagram](#) ^[1023]

5.5.5.14 Deployment Group

The Deployment group enables you to model the physical components and deployment structure of your system in a Deployment diagram.

Topics

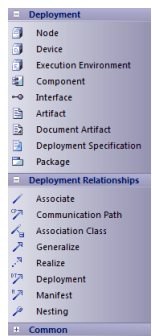
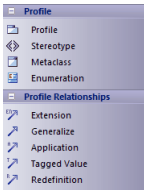
Image	Topic	Detail	See also
	Usage	<p>The Deployment group contains elements related to the actual building of the system: the components that make up the system (such as ActiveX DLLs or Java beans) and the nodes those components run on, including the physical connections between nodes.</p> <p>A Component is a piece of hardware or software that makes up the system, and a Node is a physical platform on which the component is to exist; for example, DLLs or Web Servers are Components that could be deployed on a Windows 2000 Server (Node).</p> <p>See the <i>Deployment Diagram</i> topic for an example.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram; set an element</p>	Deployment Diagram ^[1020]

Image	Topic	Detail	See also
		<p>name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	

5.5.5.15 Profile Group

The Profile group contains extended UML elements and connectors that can be used to create and modify Profiles, for rapidly creating stereotypes and Tagged Values that can be applied to structures such as elements, attributes, methods and connectors.

Topics

Image	Topic	Detail	See also
	<p>Usage</p>	<p>Profiles are used to provide a generic extension mechanism for building UML models in particular domains; they are based on additional Stereotypes and Tagged Values that are applied to structures such as elements, attributes, methods, connectors and connector ends.</p> <p>A Stereotype provides a mechanism for varying the behavior and type of a model element.</p> <p>A Metaclass is used to create a Class whose instances are Classes; a metaclass is typically used to construct metamodels.</p> <p>An Enumeration creates a Class stereotyped as enumeration, which is used to provide a list of named values as the range of a particular type.</p> <p>An Extension relationship shows that a stereotype extends a metaclass; all stereotypes must extend either one or more metaclasses, or another stereotype that extends a stereotype (that itself extends a stereotype, and so on).</p> <p>A Generalize relationship shows that one stereotype specializes a more general stereotype; the more general stereotype must still extend a metaclass.</p> <p>A Tagged Value relationship defines a reference-type (that is, RefGUID) Tagged Value owned by the source stereotype; the Tagged Value is named for the target role of this association, and is limited to referencing elements with the stereotype by the association target element.</p> <p>The Application and Redefinition relationships are deprecated.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram; set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	

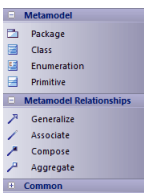
[Learn more](#)

- [Create and Modify Profiles](#) ^[1277]

5.5.5.16 Metamodel Group

The Metamodel group enables you to create metamodel diagrams with support for MOF diagrams.

Topics

Image	Topic	Detail	See also
	Usage	<p>A Package is a namespace as well as an element that can be contained in other package's namespaces.</p> <p>A Class is a representation of objects, that reflects their structure and behavior within the system.</p> <p>An Enumeration is a Class with an enumeration stereotype.</p> <p>A Primitive supports the MOF specification (deprecated - use the UML Primitive in the Class group).</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram; set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	<p>Package ^[1200]</p> <p>Class ^[1187]</p> <p>Enumeration ^[1191]</p> <p>Class Group ^[708]</p>

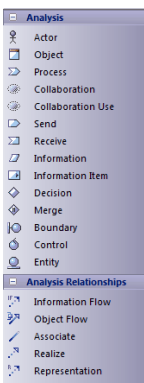
Learn more

- [MOF](#) ^[2094]

5.5.5.17 Analysis Group

Analysis-type elements are used early in modeling to capture business processes, activities, collaborations and general domain information. They are generally used in Analysis diagrams.

Topics

Image	Topic	Detail	See also
	Usage	<p>You can use stereotyped activities to model business processes, or stereotyped elements to capture standard UML business process modeling extensions such as worker, case worker, entity, and controller.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram; set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, click on the start element in the diagram and drag to the end element.</p>	

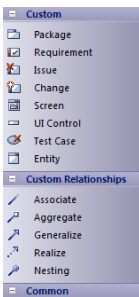
Learn more

- [Analysis Diagrams](#) ^[1509]

5.5.5.18 Custom Group

The Custom group contains a few extended UML elements that might be of use in modeling or designing your system in a Custom diagram.

Topics

Image	Topic	Detail	See also
	Usage	<p>A Package is a namespace as well as an element that can be contained in other package's namespaces.</p> <p>A Requirement is a custom element used to capture requirements outside of standard UML elements; a Requirement expresses required system behavior that can cross several Use Cases.</p> <p>You can connect Requirements to other elements using the Realize connector to express the implementation of a requirement and hence the traceability from user requirements to what is being built.</p> <p>An Issue element is a structured comment that contains information about defects and issues relating to the system/model; affected elements are connected by Trace connectors.</p> <p>A Change element is a structured comment that contains information about changes requested to the system/model; affected elements are connected by Trace connectors.</p> <p>A Screen provides a stereotyped Class element that displays a GUI type screen; this can be used to express application GUI elements and flows between them.</p> <p>A UI control likewise can be used to express GUI controls.</p> <p>A Test Case element defines what must be set up in order to test a particular feature; it enables you to define a set of tests once for a number of elements, and provides greater visibility for tests.</p> <p>An Entity is a stereotyped element that represents any general thing not captured by the element or Class type elements (for example a trading partner); use of this element is deprecated - it was originally intended to take the role now occupied by a Table element.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram; set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	<p>Package ^[1200]</p> <p>Requirements ^[1471]</p> <p>Traceability ^[635]</p> <p>Defects (Issues) ^[2244]</p> <p>Trace ^[1260]</p> <p>Changes and Defects ^[2244]</p> <p>Screen ^[1656]</p> <p>UI control ^[1657]</p> <p>Test Case ^[1677]</p> <p>Testing Workspace ^[2213]</p> <p>Table ^[1630]</p>

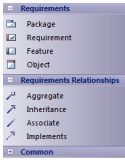
Learn more

- [Custom Diagram](#) ^[1505]

5.5.5.19 Requirement Group

As an analysis step, often it is desirable to capture simple system *requirements*. These are eventually realized by Use Cases.

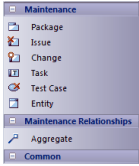
Topics

Image	Topic	Detail	See also
	Usage	<p>A Package is a namespace as well as an element that can be contained in other package's namespaces.</p> <p>Specify the Requirement of a system. Note that there are a few different requirement types, such as.</p> <ul style="list-style-type: none"> • Display • Functional • Performance • Printing • Report • Testing • Validate <p>A Feature is a small client-valued function expressed as a requirement. Features are the primary requirements-gathering artifact of the Feature-Driven Design (FDD) methodology.</p> <p>An Object is an instance of a Class.</p> <p>To add an element to the current diagram, click on the required icon, and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	<p>Package ^[1200]</p> <p>Requirement ^[1471]</p> <p>Feature ^[1485]</p> <p>Object ^[1196]</p>

5.5.5.20 Maintenance Group

The **Maintenance** elements are defects, changes, issues and tasks.

Topics

Image	Topic	Detail	See also
	<p>Usage</p>	<p>A Package is a namespace as well as an element that can be contained in other package's namespaces.</p> <p>An Issue element is a structured comment that contains information about defects and issues relating to the system/model. Affected elements are connected by Trace connectors.</p> <p>A Change element is a structured comment that contains information about changes requested to the system/model. Affected elements are connected by Trace connectors.</p> <p>A Task element defines a task attached to an element, that enables resources to be assigned specifically to the task rather than just to the parent element. You can monitor these tasks on diagram Gantt charts.</p> <p>A Test Case describes what must be set up in order to test a particular feature.</p> <p>An Entity is a stereotyped element that represents any general thing not captured by the element or Class type elements (for example a trading partner). Use of this element is deprecated: it was originally intended to take the role now occupied by a Table element.</p> <p>To add an element to the current diagram, click on the required icon, and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	<p>Package^[1200]</p> <p>Defects and Issues^[2244]</p> <p>Trace^[1260]</p> <p>Change^[2244]</p> <p>Task^[1676]</p> <p>Test Case^[1677]</p> <p>Table^[1630]</p>

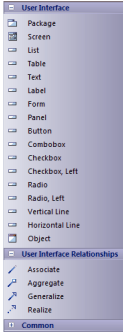

Learn more

- [Maintenance](#)^[2236]

5.5.5.21 User Interface Group

The **User Interface** group enables you to create graphical user interface diagrams.

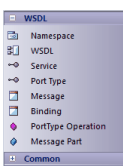
Topics

Image	Topic	Detail	See also
	Usage	<p>A Package is a namespace as well as an element that can be contained in other packages' namespaces.</p> <p>A Screen element represents a graphical user interface. You can place GUI elements onto the screen element.</p> <p>UI Control elements are placed onto the screen element to build up a graphical user interface diagram. There are different stereotyped elements such as buttons and combo boxes.</p> <p>An Object is an instance of a Class.</p> <p>To add an element to the current diagram, click on the required icon, and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	<p>Package ^[1200]</p> <p>Screen ^[1656]</p> <p>UI Control ^[1657]</p> <p>Object ^[1196]</p>
	GUI Elements	<p>The diagram on the left illustrates the GUI elements from the Toolbox, within a Screen element.</p>	

5.5.5.22 WSDL Group

The **WSDL** group gives you the ability to rapidly model and automatically generate W3C Web Service Definition Language (WSDL) documents.

Topics

Image	Topic	Detail	See also
	Usage	<p>A Namespace represents the top-level container for the WSDL model. Drag this element onto an open diagram to create the necessary model structure for WSDL documents.</p> <p>A physical WSDL document is represented as a UML component. Its interfaces represent the WSDL services.</p> <p>A WSDL Port Type is modeled as a UML interface. Its Port Type Operations are realized by Binding elements. Each of the operation parameters is derived from the Message elements defined in the Messages package.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, click on the start element in the diagram and drag to the end element.</p>	<p>WSDL Namespace ^[2069]</p> <p>WSDL document ^[2083]</p> <p>WSDL services ^[2082]</p> <p>WSDL Port Type ^[2075]</p> <p>Port Type Operations ^[2076]</p> <p>WSDL Binding ^[2078]</p> <p>WSDL Messages ^[2073]</p>

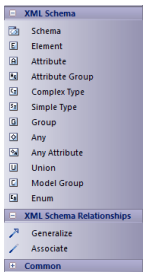
[Learn more](#)

- [Model WSDL](#) ^[2069]
- [Generate WSDL](#) ^[2085]

5.5.5.23 XML Schema Group

The **XML Schema** group provides the ability to model and automatically generate W3C XSD schema files. This group implements the constructs provided by the UML profile for XML Schema.

Topics

Image	Topic	Detail	See also
	Usage	<p>A Schema corresponds to a UML package, which contains the type and element definitions for a particular targetNamespace. Drag this item onto an open diagram to create the package to contain your schema model elements. The package is stereotyped as XSDschema.</p> <p>Open the logical diagram created under the XSDschema package and add additional schema elements as required.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram. Set the name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	

Learn more

- [Model XSD](#) ^[2027]
- [Generate XSD](#) ^[2049]
- [UML profile for XML Schema](#) ^[2055]

5.5.5.24 Data Modeling Group

This group is used for database modeling and database design, in conjunction with the *UML Data Modeling Profile*.

Topics

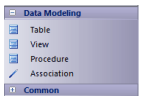
Image	Topic	Detail	See also
	Usage	<p>The Table element defines a table on the data model.</p> <p>The View element represents database views in the data model.</p> <p>The Procedure element represents stored procedures in the data model.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, click on the start element in the diagram and drag to the end element.</p>	<p>Table ^[1630]</p> <p>Views ^[1967]</p> <p>Stored Procedures ^[1967]</p>

Image	Topic	Detail	See also

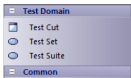
Learn more

- [Database Modeling](#)^[1625]

5.5.5.25 Test Domain Group

The Test Domain group is used to create Test Domain diagrams, used in the Testpoint facility. A Test Domain diagram provides specific elements that can aid in the logical composition of tests into Test Sets and Test Suites.

Topics

Image	Topic	Detail	See also
	Usage	<p>A <i>Test Cut</i> is a group of operations that represent a specific behavior (such as 'Print') defined for a Class, separate from other constraints defined for the Class.</p> <p>A <i>Test Set</i> aggregates one or more groups of methods (Test Cuts) that perhaps span multiple Classes, into a single task.</p> <p>A <i>Test Suite</i> aggregates one or more groups of tasks (Test Sets).</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram. Set an element name and other properties as prompted.</p>	<p>Test Cut^[2198]</p> <p>Test Set^[2198]</p> <p>Test Suite^[2199]</p>

Learn more

- [Testpoint Management](#)^[2187]

5.5.6 Diagram Tasks

This topic introduces many of the common tasks associated with managing diagrams.

Topic	Link
Add New Diagrams	Add New Diagrams ^[724]
Set Diagram Properties	Set Diagram Properties ^[725]
Add Elements and Connectors From Toolbox	Add Elements and Connectors From Toolbox ^[698]
Paste from the Project Browser	Paste from the Project Browser ^[733]

Topic	Link
Copy And Paste Diagram Element	Copy And Paste Diagram Element ^[737]
Place Related Elements on Current Diagram	Place Related Elements on Current Diagram ^[738]
Delete a Diagram	Delete a Diagram ^[738]
Rename a Diagram	Rename a Diagram ^[739]
Change Diagram Type	Change Diagram Type ^[740]
Diagram Navigation Hotkeys	Diagram Navigation Hotkeys ^[740]
Copy Image to Disk	Copy Image to Disk ^[741]
Copy Image to Clipboard	Copy Image to Clipboard ^[742]
Duplicate a Diagram	Duplicate a Diagram ^[742]
Z Order Elements	Z Order Elements ^[744]
Set Default Diagram	Set Default Diagram ^[744]
Open a Package	Open a Package ^[744]
Feature Visibility	Feature Visibility ^[745]
Insert Diagram Properties Note	Insert Diagram Properties Note ^[747]
Manage Legend Elements	Manage Legend Elements ^[748]
Autosize Elements	Autosize Elements ^[750]
Swimlanes	Swimlanes ^[751]
Swimlanes Matrix	Swimlanes Matrix ^[753]

Topic	Link
Using the Image Manager	Using the Image Manager ^[75]
Show Realized Interfaces for a Class	Show Realized Interfaces for a Class ^[76]
Label Menu Section	Label Menu Section ^[76]
Pan and Zoom a Diagram	Pan and Zoom a Diagram ^[76]
Move Diagram Sections	Move Diagram Sections ^[76]
View Last and Next Diagram	View Last and Next Diagram ^[76]
Set Diagram Page Size	Set Diagram Page Size ^[76]
Scale Image to Page Size	Scale Image to Page Size ^[76]
Lock Diagram	Lock Diagram ^[76]
Lay Out a Diagram	Lay Out a Diagram ^[78]
Undo Last Action	Undo Last Action ^[76]
Redo Last Action	Redo Last Action ^[76]
Present Diagrams in a Model Views Slideshow	Present Diagrams in a Model Views Slideshow ^[60]

Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Update Element** permission to update or delete items on a diagram, and **Manage Diagram** permission to create, copy or delete diagrams

Learn more

- [List of Available Permissions](#) ^[26]

5.5.6.1 Add New Diagrams

This topic explains how to add a UML diagram, Extended diagram or MDG Technology diagram to a model in Enterprise Architect.

How to

Step	Action	See also
1	In the Project Browser, select the appropriate package or element under which to place the diagram.	
2	<p>Do one of the following:</p> <ul style="list-style-type: none"> In the Project Browser toolbar click on the New Diagram icon Right-click to open the context menu and select the Add Add Diagram or Add Add <type> Diagram menu option Press (Insert) and select the Add Add Diagram or Add Add <type> Diagram menu option, or Select the Project New Diagram menu option <p>The New Diagram dialog displays.</p>	
3	The Name field defaults to the name of the selected package or element; if necessary, type a different name for the new diagram.	
4	<p>In the Select From panel, click on the appropriate diagram category for the diagram.</p> <p>The Diagram Types panel displays a list of the diagram types within the selected category.</p>	UML Diagrams ¹⁰¹⁰
5	In the Diagram Types panel, click on the type of diagram to create.	
6	<p>Click on the OK button to create your new diagram.</p> <p>Enterprise Architect checks whether there is a template diagram for this diagram type in the Templates package and, if so, applies the display characteristics defined in that template.</p>	Set Element Templates Package ⁸²⁰

Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Manage Diagrams permission to create new

diagrams

- The diagram type determines the default toolbar associated with the diagram and whether it can be set as a child of another element in the Project Browser (for example, a Sequence diagram under a Use Case)

Learn more

- [Manage Diagrams](#) ^[264]

5.5.6.2 Diagram Properties

You can set several properties of a diagram using the diagram Properties dialog. Some properties influence the display and some are logical attributes that appear in the documentation.

Topics

Topic	Detail	See also
Usage	<p>There are several options for opening the diagram Properties dialog for a given diagram:</p> <ul style="list-style-type: none"> • Select the Diagram Properties menu option to open the Properties dialog for the currently active diagram • Right-click on the required diagram in the Project Browser and select the Properties context menu option • Right-click on the background of the open diagram and select the Properties context menu option • Double-click in the background of the open diagram <p>In the Diagram Properties dialog you can set properties including name, author and version information, zoom factor, paper size and layout, diagram notes and various appearance attributes.</p> <p>Once you have made any necessary changes, click on the OK button to save the data and exit the dialog.</p>	

Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Update Diagrams** permission to change diagram properties
- You can also set the default diagram background color and the element fill color on the Standard Colors page of the Options dialog; you can set color gradients for both diagram background and element fill color on the Diagram Appearance page of the dialog

Learn more

- [Update Diagrams](#) ^[264]

- [Standard Colors](#) ^[546]
- [Diagram Appearance](#) ^[550]
- [General Tab](#) ^[726]
- [Diagram Tab](#) ^[727]
- [Elements Tab](#) ^[729]
- [Features Tab](#) ^[731]
- [Connectors Tab](#) ^[732]

5.5.6.2.1 General Tab

The General tab of the diagram Properties dialog enables you to define characteristics of the overall diagram, such as its title, version and modification date.

Access [Diagram | Properties > General](#)

Use to

- Define characteristics of the overall diagram

Reference

Field	Usage	See also
Name	Indicates the name of the diagram (defaults to the name of the parent package).	
Author	Specifies the name of the person who owns the diagram (defaults to the name of the person who created the diagram).	
Version	Indicates the version number of the diagram (defaults to 1.0).	
Stereotype	Indicates the name of the stereotype for the diagram. You can define stereotypes to select here using the Settings UML Types menu option, selecting the Stereotypes ^[2580] tab and creating stereotypes with a Base Class of Diagram .	Stereotypes Tab ^[1293]
Created	Automatically display the date the diagram was created.	
Modified	Indicates the date and time on which the diagram was last modified (defaults to the current date and time).	
Notes	Indicates any additional notes about the diagram. You can format the notes using the Notes toolbar at the top of the field.	Notes Toolbar ^[979]

Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Update Diagrams permission to update diagram properties

Learn more

- [Permissions List](#)^[264]

5.5.6.2.2 Diagram Tab

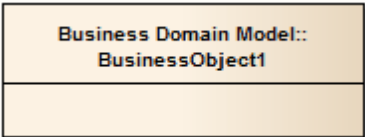
The Diagram tab of the diagram Properties dialog enables you to define the representation of the diagram.

Access [Diagram | Properties > Diagram](#)

Use to

- Define the representation of a diagram

Reference

Field	Usage	See also
Use Alias if Available	Display the element alias as the name if the alias is specified.	
Show Additional Parents	Show the name of all parents not in the current diagram for all Classes and interfaces.	
Show Page Border	Show a page border to align elements with.	
Show Diagram Details	Show diagram details in a note in the top left corner of the diagram. Deselect the option to hide the diagram details.	
Show Sequence Notes	Show the Sequence Notes on the current diagram.	
Show Namespace	Show the namespace of each element on the diagram, as part of the element name 	

Field	Usage	See also
<p>Fully Qualified Namespace</p>	<p>(If Show Namespace is selected) show the fully qualified namespace of each element on the diagram, under the element</p> <div data-bbox="491 506 863 667" data-label="Diagram"> <p>The diagram shows a rectangular box with a light tan fill and a black border. Inside the box, the text "BusinessObject1" is centered. Below the box, the text "(from Business Rules Model::Business Domain Model)" is displayed in a smaller font.</p> </div>	
<p>Hand Drawn</p>	<p>Display the diagram contents as if they had been roughly drawn by hand</p> <div data-bbox="485 869 995 1151" data-label="Diagram"> <p>The diagram is a sequence diagram drawn in a hand-drawn style. It features two rectangular boxes at the top labeled "Object1" and "Object2". Below each box is a vertical line representing its lifeline. A solid arrow points from the lifeline of Object1 to the lifeline of Object2, labeled "Message()". A dashed arrow points from the lifeline of Object2 back to the lifeline of Object1, labeled "Return Message()".</p> </div>	
<p>Whiteboard Mode</p>	<p>Displays the diagram as if it had been drawn on a whiteboard, with white fill and background</p> <div data-bbox="485 1350 995 1655" data-label="Diagram"> <p>The diagram is a sequence diagram in whiteboard mode. It features two rectangular boxes at the top labeled "Object1" and "Object2". Below each box is a vertical line representing its lifeline. A solid arrow points from the lifeline of Object1 to the lifeline of Object2, labeled "Message()". A dashed arrow points from the lifeline of Object2 back to the lifeline of Object1, labeled "Return Message()".</p> </div>	
<p>Disable fully scoped object names</p>	<p>Prevent the display of fully-qualified object names on the diagram.</p>	
<p>Page Setup</p>	<p>Adjust the scaling of the image according to the size of the page it is to be printed on.</p>	<p>Scale Image to Page Size^[766]</p>

Field	Usage	See also
Print Page Header and Page Footer	Add page headers and footers to a print-out of the diagram; the headers and footers are generated from the diagram characteristics, such as the name of the creator and the date of modification.	
Always Open as Element List	Always display the diagram contents as a Diagram List rather than as a diagram.	Diagram List ^[596]
Always Open as Gantt	Always display the diagram contents as a Gantt chart.	
Set Layout Style	Define the layout of the diagram, to be applied automatically when you select the Layout Diagram options.	Layout Diagram ^[787]
RTF Document Options	Options for generating RTF reports for a particular diagram.	Document Options ^[2255]
Exclude image from RTF documents	Exclude this diagram image from any RTF document generated on the parent package or element.	
Document each contained element in RTF	Include documentation on each element in this diagram, in any RTF document generated on the parent package or element.	Generate RTF Documentation ^[2255]
Divide Diagram into Multiple Pages	Divide each large diagram into separate pages in the RTF document.	Scaled to page ^[766]
Rotate Images	Rotate each diagram image by 90 degrees in the RTF document.	

Notes

- The **Divide Diagrams into Multiple Pages** option is only effective when the **Scaled Printing** option is set to **None** on the Print Advanced dialog
- **Rotate Images** is only available for bitmap (.bmp) images

5.5.6.2.3 Elements Tab

The Elements tab of the diagram Properties dialog enables you to define what components of the elements should be displayed on the diagram.

Access **Diagram | Properties > Elements**

Use to

- Define what components of elements should be displayed

Reference

Field	Usage	See also
Use Stereotype Icons	<p>For elements that have whole shapes drawn by Enterprise Architect (such as Analysis stereotypes), draw the alternative shape (if defined).</p> <p>For elements that have an icon displayed in the top right corner, (such as an Artifact element) if Show Element Stereotypes is selected, display the stereotype icon instead of the stereotype text.</p>	Analysis stereotypes ^[1508] Artifact ^[1179]
Show Element Stereotypes	<p>For elements that have whole shapes drawn by Enterprise Architect, if Use Stereotype Icons is deselected, display any stereotype on the element.</p> <p>For elements that have an icon displayed in the top right corner, indicate that a stereotype is present (icon if Use Stereotype Icons is selected, text if not).</p>	
Show Table Owner	Display the Table Owner.	Set Table Owner ^[1939]
Show Element Property String	Show the advanced property string for all elements; for example, {leaf}.	
Show Compartments	<p>Enable the following compartments to be shown or hidden for any element using rectangle notation:</p> <ul style="list-style-type: none"> • Attributes • Operations • Tags (Tagged Values) • Requirements • Constraints • Testing (Testing Scripts) • Maintenance (Maintenance Scripts) • Package Contents • Notes 	Show Test Scripts Compartment ^[2229] Show Maintenance Script in Diagram ^[2242]

5.5.6.2.4 Features Tab

The Features tab of the diagram Properties dialog enables you to define how features (attributes and operations) are displayed on the diagram.

Access [Diagram | Properties > Features](#)

Use to

- Define how features (attributes and operations) are displayed on a diagram

Reference

Field	Usage	See also
Show Qualifiers and Visibility Indicators	Show or hide the qualifiers and visibility indicators on the diagram. <i>Qualifiers</i> include such things as the 'derived' symbol (<i>l</i>) and the public key symbol (PK). Visibility indicators ^[174] include such things as +, -, # and ~, which indicate the scope of access of the item (such as an attribute, operation or role).	Visibility indicators ^[174]
Show Stereotypes	Show the stereotypes on all features.	
Show Property String	Show the advanced property string for all element features, for example, {readOnly}.	
Show Operation Return Type	Display the return data type of operations.	
Suppress Brackets for Operations Without Parameters	Suppress brackets on operations that have no parameters; that is, Opn ; rather than Opn() ;	
Always Show Linked Features	Force display of linked ^[934] attributes and operations, regardless of the setting of any other option that might hide them.	Linked Attributes and Operations ^[944]
Visible Class Members	Hide Class members according to their scope and methods that specify properties.	Visible Class Members ^[732]
Show Attribute Detail	Select whether to show both the attribute name and type, or the attribute name only.	

Field	Usage	See also
Show Parameter Detail	Control the display of method parameters.	Visible Class Members ^[732]

5.5.6.2.4.1 Visible Class Members

On the Features tab of the diagram Properties dialog, the Visible Class Members panel enables you to hide Class members by their scope and methods that specify properties. Use the check boxes to define the visibility of Class members.

Access [Diagram | Properties > Features : Visible Class Members](#)

Use to

- Define visibility of class members

Show Parameter Detail:

The **Show Parameter Detail** field enables you to control the display of method parameters with the following options:

Option	Effect
None	No details shown.
Type Only	Show the type of parameter only.
Full Details	Show all of the details for parameters.
Name Only	Show the name of the parameter only.

5.5.6.2.5 Connectors Tab

The Connectors tab of the diagram Properties dialog enables you to define the appearance of the connectors on the diagram.

Access [Diagram | Properties > Connectors](#)

Use to

- Define the appearance of connectors on a diagram

Reference

Field	Action	See also
Show Relationships	Show relationships in the current diagram.	
Show Collaboration Numbers	Show numbering in Communication diagrams.	
Show Non-Navigable Ends	Indicate when an Association end is not navigable; a cross is presented at the Association connector.	
Show Connector Property String	Show the property string for connectors.	
Suppress All Connector Labels	Hide all connector labels.	
Show Stereotype Labels	Show (selected) or hide (deselected) any stereotype labels on the connectors on the diagram.	
Connector Notation	Display the required connector notation: <ul style="list-style-type: none"> • UML 2.1 - use the standard UML 2.1 notation for connectors • Information Engineering - use the Information Engineering (IE) connection style • IDEFX1 - use the Integrated Definition Methods IDEFX1 connection style 	http://www.agiledata.org/essays/dataModeling101 http://www.edef.com/IDEF1X.html

5.5.6.3 Paste from Project Browser

As well as creating new elements in a diagram, you can drag existing elements from the Project Browser into the current diagram.

When you drag an existing element into the diagram, the Paste Element dialog displays to prompt you to select the type of paste action to carry out. (If the dialog does not display, press and hold (**Ctrl**) while dragging the element onto the diagram.)

Topics

Topic	Detail	See also
Pasting Options	Three paste options are available: <ul style="list-style-type: none"> • Paste the element as a simple link: in this case the original 	

Topic	Detail	See also
	<p>element exists both in the current diagram and in the original source diagram; changes to the element are reflected in all diagrams in which it is shown</p> <ul style="list-style-type: none"> • Paste as an instance of the element: if the element can have instances such as an Object, Sequence instance or Node instance, you can drop the element in as an instance of the source element, with the classifier pre-set to the original source; this is useful when creating multiple instances of a Class in a Sequence diagram or Communication diagram <p>For a State Machine element, this option is replaced by the option to paste as an invocation of the State Machine; the resulting element is a State object with the State Machine as its classifier</p> <p>When you select the instance or invocation options, the Copy connectors checkbox is enabled; if you select this checkbox, any connectors between the original element and any other elements that have also been pasted to this diagram are reproduced as connectors between the instances</p> <ul style="list-style-type: none"> • Create as a child of the source element: this automatically creates a new Class (which you are prompted to name) with a Generalization connector back to the source; this is very useful when you have a Class library or framework from which you inherit new forms - for example, you can paste a Hashtable as "MyHashtable" which automatically becomes a child of the original Hashtable <p>Used with the Override parent operations feature, this is a quick way to create new structures based on frameworks such as the Java SDK and the .NET SDK</p> <p>You can make your selection on this dialog the default for:</p> <ul style="list-style-type: none"> • All drag and drop operations, or • Only those where you display this Paste Element dialog <p>If you select the This Dialog checkbox, you should then select the Only</p>	<p>Object Classifiers [898]</p> <p>Override Parent Operations [908]</p> <p>Diagram Behavior [553]</p>

Topic	Detail	See also
	<p>show this dialog when (Ctrl) +Mouse drag is used checkbox and, on the Diagram Behavior page of the Options dialog, the Auto Instance checkbox.</p> <p>The effect of these selections is to give you two default paste options:</p> <ul style="list-style-type: none"> • Just drag the element onto the diagram and automatically create an instance • Press (Ctrl) while you drag the element from the Project Browser, displaying the Paste Element dialog, and click on the OK button to automatically paste the element according to whatever option you last selected from the dialog <p>If you select the All Drag and Drop checkbox on the Paste Element dialog, this deselects the Auto Instance checkbox on the Options dialog and enables you to add existing elements to the diagram according to the paste option you selected, without pressing (Ctrl) and without displaying a dialog.</p> <p>If you want to change the default paste option, press (Ctrl) as you drag to display the dialog again and make your changes.</p>	

Notes

- When an element dragged from the Project Browser is pasted into the diagram, its default style and size is applied rather than any alternative styles and sizes used for instances of the element in other diagrams

Learn more

- [Connect Requirements](#)^[1480]
- [Create Object From Attribute](#)^[896]
- [Make Linked Element A Local Copy](#)^[823]

5.5.6.3.1 Paste Multiple Items

You can paste multiple elements from the Project Browser into the current diagram.

Topics

Topic	Detail	See also
Pasting Multiple Elements	<p>To select multiple elements, click on the selected items from the Project Browser while pressing and holding:</p> <ul style="list-style-type: none"> • (Ctrl) to add single items to the selection of multiple elements, or 	Pasting Options ^[733]

Topic	Detail	See also
	<ul style="list-style-type: none"> • (Shift) to select all the elements between the first and last selected items in the Project Browser <p>You can then drag the selected elements from the Project Browser onto the current diagram, pressing and holding (Ctrl); for each element you have selected, the Paste Element dialog displays, prompting you to select the type of paste action to carry out.</p>	

5.5.6.3.2 Paste Composite Elements

When you drag a Composite element from the Project Browser onto the current diagram, Enterprise Architect prompts you to select the type of paste action to carry out with the Composite element.

(If the dialog does not display, press and hold (**Ctrl**) while dragging the element onto the diagram.)

Topics

Topic	Detail	See also
Pasting composite elements	<p>Two advanced options are available for pasting Composite elements; these require the include Embedded Elements checkbox to be selected:</p> <ol style="list-style-type: none"> 1. The All Embedded Elements option, which pastes all of the Composite element's embedded elements. 2. The Based on instance option, which pastes only the elements contained in a specific instance of the Composite element, maintaining the layout of the embedded elements. <p>Click on the drop-down arrow and select the appropriate instance.</p> <p>For details of the other options on this dialog, see the <i>Paste from Project Browser</i> topic.</p>	Paste from Project Browser ^[733]

5.5.6.3.3 Paste Activities

You can paste an Activity from the Project Browser into the current diagram.

Topics

Topic	Detail	See also
Pasting Activities	When you drag an Activity from the Project Browser onto the current diagram, The Paste Element dialog displays, prompting you to select the type of paste action to carry out. (If the dialog	Paste from Project Browser ^[733]

Topic	Detail	See also
	<p>does not display, press and hold (Ctrl) while dragging the element onto the diagram.)</p> <p>Two options are available:</p> <ul style="list-style-type: none"> • Paste the Activity as a link: in this case the Activity appears in the current diagram as a simple reference to the original source Activity. Changes to the Activity in the diagram affect all other links to this Activity • Paste as an invocation of the Activity; if you select this option, the Copy connectors checkbox is enabled. If you select this checkbox, any connectors between the original Activity and any other elements that have also been pasted to this diagram are reproduced as connectors between the instances <p>For details of the other options on this dialog, see the <i>Paste from Project Browser</i> topic.</p>	

Learn more

- [Activity](#)^[10]

5.5.6.4 Copy And Paste Diagram Element

How to

To copy a diagram element

Step	Instruction	See also
1	Select the element(s) to copy.	
2	For multiple elements, right-click to open the context menu and select the Copy menu option. Alternatively, press (Ctrl+C).	
3	For single elements, select the Edit Copy menu option or alternatively press (Ctrl+C).	

To paste diagram elements

Step	Instruction	See also
1	Open the diagram to paste into.	
2	Right-click on the diagram background to open the diagram context menu.	
3	Select either the Paste Object(s) as New menu option (completely new element) or the Paste Object(s) as Link menu option (reference to the existing element).	

Notes

- The **Date Created** and **Time Created** parameters of a pasted-as-new element are set to the current date and time; the parameters for a linked element remain the same as the copied element

5.5.6.5 Place Related Elements on Diagram

To find and place related elements on the current diagram, use the Relationships window.

Access [View | Relationships](#)

Topics

Topic	Detail	See also
Usage	<p>Right-click on any connector in the list to open the context menu.</p> <p>If an element is not present in the current diagram, the context menu contains the Place Target Element in Diagram option. This is useful when you are building up a picture of what an element interacts with, especially when reverse engineering an existing code base.</p> <p>Select the Place Target Element in Diagram option.</p> <p>Move the cursor to the required position in the diagram and click to place the element. Alternatively, press (Esc) to cancel the action.</p>	

5.5.6.6 Delete Diagram

How to

To delete a diagram from your model

Step	Action	See also
1	In the Project Browser, right-click on the diagram to delete. The context menu displays.	

Step	Action	See also
2	Select the Delete '<diagram name>' menu option. A confirmation prompt displays.	
3	Click on the OK button to confirm the delete.	

Notes

- In Enterprise Architect there is no Undo feature for deleting diagrams, so be certain that you want to delete a diagram before you do so
- When you delete a diagram, you do not delete the elements in the diagram from the model
- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Diagrams** permission to create new diagrams
- You can also delete multiple diagrams from the Project Browser, by holding (**Ctrl**) or (**Shift**) while you select them, then right-clicking on one of them and selecting the **Delete selected items** context menu option

Learn more

- [Permissions List](#)^[264]

5.5.6.7 Rename Diagram

How to

To rename a diagram

Step	Action	See also
1	Open the Diagram Properties dialog by double-clicking on the diagram background, or by selecting the Diagram Properties menu option.	
2	In the Name field on the General tab, type the new name for your diagram.	
3	Click on the OK button to save changes.	

5.5.6.8 Change Diagram Type

If necessary, you can change one type of diagram to another type. This is useful if you have either made a mistake in selecting the diagram type to begin with, or if the purpose and nature of a diagram changes during analysis.

Access [Diagram](#) | [Advanced](#) | [Change Type](#)

How to

To change a diagram type

Step	Action	See also
1	Open the diagram to change.	
2	Select the Change Type menu option. The Change Diagram Type dialog displays.	
3	Select the required diagram type.	
4	Click on the OK button to save changes.	

Notes

- Some diagram types do not transfer to others; for example you cannot change a Class diagram into a Sequence diagram

5.5.6.9 Diagram Navigation Hotkeys

The diagram hotkeys enable you to quickly navigate to and select elements within a diagram. The following table details the key combinations and their functionality.

Hotkey Command	Action
Shift + Arrow , Element(s) selected	Move the selected element(s) by increments.
Arrow , No element selected	Scroll around the diagram.
Esc	Clear the current selection.
Tab	Select the first element in the diagram if none currently selected.

Hotkey Command	Action
Shift +click	Add the clicked element to the current selection.
Ctrl +click	Add the clicked element to the current selection.
Ctrl + Shift +drag	Pan the diagram.
Alt + G	Select the item in the Project Browser and give it focus.

5.5.6.10 Copy Image to Disk

You can copy a diagram image to a disk file in the following formats:

- Windows bitmap (256 color bitmap)
- GIF image
- Windows Enhanced Metafile (standard metafile)
- Windows Placeable Metafile (older style metafile)
- PNG format
- JPG
- TGA

How to

To copy a diagram image to file

Step	Action	See also
1	Open the diagram to save.	
2	Select the Diagram Save as Image menu option, or press (Ctrl+T).	
3	When prompted, enter a name for the file and select an image format.	
4	Click on the OK button.	

Notes

- Enterprise Architect clips the image size to the smallest bounding rectangle that encompasses all diagram elements

5.5.6.11 Copy Image to Clipboard

You can copy diagram images onto the MS Windows clipboard and paste them directly into MS Word or other applications.

How to

To copy an image to the clipboard

Step	Action	See also
1	Open the diagram to copy.	
2	Select the Diagram Copy Image to Clipboard menu option, or press (Ctrl+B).	
3	Click on the OK button. The diagram has been copied to the clipboard and can now be pasted into compatible applications or into another diagram. You can set the clipboard format on the Options dialog (Tools Options General). Enterprise Architect supports bitmap or metafile format.	General Settings ⁵⁴⁴

5.5.6.12 Copy (Duplicate) Diagram

Enterprise Architect makes it easy to duplicate a complete diagram, with either :

- links back to the original diagram elements (*shallow mode*) or
- complete copies of all elements in the diagram (*deep mode*) or
- complete copies of all elements in the diagram that have the same parent as that of the diagram and links back to the original diagram elements for all other elements on the diagram (*smart mode*)

How to

To duplicate a diagram

Step	Action	See also
1	In the Project Browser, select the diagram to copy.	
2	Right-click to display the context menu and select the Copy Diagram menu option.	
3	Navigate to the package to host the new diagram, and right-click to open the context menu.	

Step	Action	See also
4	Select the Paste Diagram menu option. The Copy Diagram dialog displays.	
5	In the Name field, type the name for the new diagram.	
6	In the Type of copy panel, click on the radio button for the type of copy you require; either: <ul style="list-style-type: none"> • Shallow - Link elements • Deep - Complete copies of the original elements shown on the diagram (except Packages), or • Smart - Complete copies of the original elements shown on the diagram that have the same parent as that of the diagram and linked elements of all other elements on the diagram 	
7	Click on the OK button. Enterprise Architect automatically creates the new diagram, links or creates new elements, and arranges them as in the original diagram. All links are also copied between diagram elements where appropriate.	

Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Diagrams** permission to copy diagrams
- You can also paste a copied diagram as a child of a composite element
- For a diagram copied in *shallow mode*, any changes to the properties of the original elements are reflected in the elements in the pasted diagram
- For a diagram copied in *deep mode*, any changes to the properties of the original elements are not reflected in the elements in the pasted diagram
- For a diagram copied in *deep mode*, package elements are copied as simple links only; you cannot paste as new
- For a diagram copied in *smart mode*, any changes to the properties of the original elements are reflected only in those elements that are copied as simple links in the pasted diagram
- Element position and size are independent in all copy modes

Learn more

- [Permissions List](#)^[264]

- [Composite Elements](#) ^[825]

5.5.6.13 Z Order Elements

Z Order refers to an element's depth in the diagram perspective, and thus influences which elements appear in front of others and which appear behind.

How to

To set the Z Order of an element

Step	Action	See also
1	Right-click on the element in the diagram.	
2	Select the Z-Order menu option.	
3	The displayed commands enable you to move the selected element(s) back, forward, to the front of all other elements or behind all other elements.	

Notes

- This option enables you to change the Z-order of a single element approximately in relation to the rest of the diagram; you can also select the **Modify Z-Order** option from the *diagram* context menu, to review and adjust the Z-Order of each element on the diagram precisely relative to each other element

Learn more

- [Diagram Context Menu](#) ^[687]

5.5.6.14 Set the Default Diagram

A project might have a default diagram. If set, this diagram loads when Enterprise Architect first opens the model. It is often convenient to place hyperlinks to other diagrams and resources on the default diagram, thus creating a Home Page for your model.

Access **Diagram | Advanced | Make Model Default**

Notes

- Once you have specified a default diagram, the **Home** icon on the Diagram toolbar takes you back to that diagram from your current location in the model

5.5.6.15 Open Package From Diagram

How to

To open a package from within a diagram follow the steps below:

Step	Action	See also
1	Open a diagram that shows the package to open.	
2	Right-click on the package element to open the context menu.	
3	Select the Open Package option. Alternatively, press (Ctrl+K).	

Notes

- Enterprise Architect finds the package default diagram and opens it for you; this is the first available diagram in the package, selected in alphabetical order - for example, a diagram called Alpha in a child package or element several levels down opens before a diagram called Beta immediately under the selected package

5.5.6.16 Feature Visibility

Enterprise Architect enables you to set the visibility of attributes and operations - where shown - for selected elements on a specific diagram only. You can hide or show attributes and operations by scope, or you can hide attributes and operations individually. The visibility you set applies only to the current diagram, so a Class can appear in one diagram with all features displayed, and in another with features hidden. For example, you can hide all protected attributes, all private operations or any other combination of attributes and operations.

It is possible to show inherited attributes, operations, requirements, constraints and Tagged Values for elements that support those features. When Enterprise Architect displays inherited features, it creates a merged list from all generalized parents and from all realized interfaces. If a child Class redefines something found in a parent, the parent feature is omitted from the Merge List.

How to

To customize feature visibility

Step	Action	See also
1	<p>Either:</p> <ul style="list-style-type: none"> Click on the element in the diagram and either click on the Element Feature Visibility menu option or press (Ctrl+Shift+Y), or Right-click on the element in the diagram to display the context menu and click on the Feature Visibility option <p>The Feature Visibility dialog displays.</p>	
2	To filter display of attributes or operations by scope, select the checkbox against each scope that should be visible and clear the checkbox against each scope that	

Step	Action	See also
	should not.	
3	<p>In the Show Element Compartments panel, select the compartments to display for the element on the diagram.</p> <p>The Fully Qualified Tags checkbox enables you to display the full provenance of a Tagged Value, where the same Tagged Value can be used several times in different contexts with different values. The description in the Tagged Value compartment reads:</p> <p style="text-align: center;"><Profile>::<Stereotype>::<Tagged Value name>=<Value> for example: BPMN::Activity::Activity Type = Task.</p> <p>(Only for Tagged Values created in Enterprise Architect release 7.1 or later.)</p> <p>If you select the Notes checkbox, the Notes compartment on the element in the diagram displays the text that has been typed into the Notes field of the Element Properties dialog.</p> <p>This checkbox also enables the maximum chars field, which defaults to 1000 as the number of characters of notes text displayed. Overtyping this value to display less text or more text.</p> <p>The change only applies to the selected elements on the diagram, so you can display full notes for a selected element whilst the other elements on the diagram have no notes text.</p>	
4	<p>In the When Resizing Elements panel, select the appropriate option for resizing the Class, object or table to prevent very wide diagram objects.</p> <p>The selected option defaults to Resize to longest Feature, so that the minimum width for a diagram object is determined by its longest displayed attribute, method or other compartment value.</p> <p>If necessary, you can change the option to Wrap Features (so that any longer features are wrapped onto multiple lines) or Truncate Features (so that longer features are not displayed in full).</p>	
5	<p>If required, in the Inherited Features panel, select one or both checkboxes to set whether Enterprise Architect should display inherited features as well as directly owned ones.</p>	
6	<p>Click on the OK button to save changes.</p> <p>Enterprise Architect redraws the diagram with the appropriate level of feature visibility.</p>	

Topics

Topic	Detail	See also
Suppress or Show Specific Features	<p>The Custom button in the Attribute Visibility and Operation Visibility panels enables you to show or hide specific operations and attributes.</p> <p>If you select the Show checkbox, the Custom button displays the Show Features in Diagram dialog; if you deselect the checkbox, the button displays the Suppress Features in Diagram dialog.</p> <p>The two dialogs are identical, but in the first you select the checkboxes of specific features to <i>show</i>, and in the second you select the checkboxes of specific features to <i>hide</i>.</p> <p>You can also use the Filter by Scope button in this dialog to, for example, list only operations that are Protected and select, say, two of them to hide, so that on the diagram the element displays all but two of the Protected operations and all operations of other scopes.</p>	

Notes

- To show features for element types that do not have visible compartments, such as Use Cases and Actors, right-click on the diagram object to display the context menu and select the **Advanced Settings | Use Rectangle Notation** option
- The **Show** checkbox, if selected, overrides these selections to display all attributes or operations in the element, except those specifically deselected in the Show Features in Diagram dialog
- If you have selected the **Notes** checkbox, you can select the **Render Formatted Notes** checkbox to display the text on the diagram, formatted using the Notes toolbar

Learn more


- [Notes](#) ⁹⁷⁹

5.5.6.17 Insert Diagram Properties Note

Properties of a diagram can be displayed on screen within a custom text box. You can move this text box around and change its appearance. You cannot change what the text box says.

Topics

Topic	Detail	See also
Usage	<p>To create the note, drag the Diagram Notes icon from the Common page of the Toolbox onto the diagram.</p> <p>Alternatively:</p> <ul style="list-style-type: none"> • Select the Diagram Property Note menu option, or • Click on the Diagram Properties Note button on the UML 	

Topic	Detail	See also
	<p>Elements toolbar and click on the diagram</p> 	

Notes

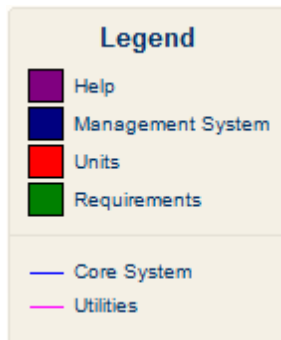
- This is not the same as the diagram details note, which displays in the top left corner of the diagram if the **Show Diagram Details** checkbox is selected on the Diagram Properties dialog; you cannot move the diagram details, nor change the appearance - to hide the diagram details, deselect the checkbox

Learn more


- [Configure Default Appearance](#)^[81]
- [Diagram Properties](#)^[72]

5.5.6.18 Create Legends

A *Legend* shape identifies colors and styles you have used to group other elements on the diagram. You can use the Legend to assist in distinguishing different elements, connectors or systems on the diagram. For example, the Legend could show that all elements concerned with the management system are shaded in blue, and all outcomes connectors are shown in red. The Legend displays as a key to the diagram, with the filled shape styles first and the lines and connector styles underneath.



You add a Legend to the diagram, then edit it to add Legend elements, which define the colors and styles used in the diagram.

To add a Legend to a diagram, drag the *Diagram Legend* element from the Common page of the Toolbox onto the diagram (or click on the **New Diagram Legend** icon () on the UML Elements toolbar, and click on the diagram).

To edit the Legend either double-click on the Legend or Right-click on the Legend and select the **Properties** context menu option.

5.5.6.18.1 Legend Properties

Access Drag the Diagram Legend element from the Common page of the Toolbox onto the diagram

Use to

- Add a Legend to a diagram

Reference

Field	Usage	See also
Name	Indicates a name for an element displayed within a legend.	
Fill Color	Indicates a fill color for an element within a legend.	
Line Color	Indicates a line color for an element within a legend.	
Line Thickness	Denotes the line thickness of an element within a legend.	
Style Options	Allows further alteration options to the legend appearance.	Style Options ^[750]
Fill	Indicates that a fill element is to be created/modified.	
Line	Indicates that a line element is to be created/modified.	
Hand (Up)	If more than one element is present within a legend, this button moves a highlighted element up one entry within the description list.	
Hand (Down)	If more than one element is present within a legend, this button moves the highlighted element down one entry within the description list.	
New	Add a new element to the legend.	
Save	Saves an element and adds it to the description list.	
Delete	Deletes an element from the description list.	

Description List	Specifies the current elements created within the legend.	
-------------------------	---	--

Notes

- The Legend dialog enables you to add, delete, modify or re-sequence Legend elements; use the Fill tab to define the Legend elements for shapes, then click on the Line tab to define Legend elements for lines and connectors

5.5.6.18.2 Style Options

Click on the **Style Options** button (...) to display the Style Options dialog, on which you can modify a Legend title, font size, background color and border color. If you choose default options for the colors, the Legend automatically assumes colors based on the diagram background color.

Access Legend dialog > (...)

Use to

- Modify appearance options for a Legend

Reference

Field	Usage	See also
Legend Title	Indicates the title of the Legend.	
Heading Size	Indicates the size of the font for a Legend.	
Font Color	Indicates the font color the Legend is to use.	
Background	Indicates the background color the Legend is to use.	
Border Color	Indicates the border color the Legend is to use.	

5.5.6.19 Autosize Elements

You can autosize an element or group of elements in a diagram to the default size for the element type (for a Class, 90 x 70 pixels at 100% zoom). However, if the element contains more information than the default size can show (such as a long name, long attributes or additional compartments) the autosize option resizes the element to the minimum size for revealing the information.

The size change effectively operates around the mid point of each element, so the layout and size of the diagram do not change. Automatically changing the layout of a diagram is a different process.

How to

To autosize elements

Step	Action	See also
1	Select the elements to resize; (press (Ctrl+A) to select all).	
2	Either: <ul style="list-style-type: none"> • Right-click on any of the elements and, on the context menu, select the Autosize menu option, or • Press (Alt+Z). 	

Notes

- Not all elements resize: elements such as Events remain the same; Timing and Sequence diagrams (where position is crucial) are unchanged; and elements added from a profile or Shape Script maintain any size definitions imposed by the profile
- With an element image created with a Shape Script that contains a defSize command, **Autosize** returns the element to the defSize value and not the element default size

Learn more

- [Lay Out a Diagram Automatically](#)^[787]
- [defSize command](#)^[1368]

5.5.6.20 Swimlanes

Enterprise Architect diagrams support *Swimlanes* for all diagram types. Swimlanes are vertical or horizontal bands in a diagram that divide the diagram into logical areas or partitions. In the example below the activities relating to particular entities within the model (such as the User, or the back end Repository) are placed within a containing swim lane to indicate their association.



Access [Diagram | Swimlanes and Matrix > Swimlanes](#)

Use to

- Add Swimlanes to a diagram

Reference

Field	Usage	See also
Active	Indicates that the Swimlane dialog is active, as opposed to the Matrix dialog.	Swimlanes Matrix ^[753]

Orientation	Specifies the orientation of the swimlane, either vertical or horizontal.	
Line Color	Indicates the line color for the swimlane.	
Font Color	Indicates the font color for the swimlane.	
Line Width	Indicates the line width of the swimlane.	
Locked	Indicates that the swimlane can not be moved on the diagram.	
Bold Font	Indicates that the name given to the swimlane is set to bold.	
Hide Classifier	Indicates that the classifier name is not shown on the diagram.	
Hide Names	Indicates that the names set for swimlanes do not appear on the dialog.	
Show Names in Title Bar	Specifies that the name given to a swimlane appear in the title bar.	
Hand (Up)	If more than one element is present within a swimlane, this button moves a highlighted element up one entry within the Swimlanes list.	
Hand (Down)	If more than one element is present within a swimlane, this button moves a highlighted element down one entry within the Swimlanes list.	
New	Opens the Swimlane Details dialog, allowing specification of the Name, Classifier and Background Color.	Swimlane Details 
Modify	Opens the Swimlane Details dialog, allowing modification of the Name, Classifier and Background Color.	Swimlane Details 
Delete	Deletes a Swimlane from the Swimlanes List.	

Swimlanes List	The list of the currently defined Swimlanes.	
-----------------------	--	--

Notes

- If you set a background color for a swimlane, it takes on the same shading profile as the main diagram background

5.5.6.20.1 *Swimlane Details*

The Swimlane Details dialog opens if the **New** or **Modify** buttons are clicked from the Swimlanes dialog.

Access [Diagram | Swimlanes and Matrix > Swimlanes : New/Modify](#)

Use to

- Specify details about a swimlane

Reference

Field	Usage	See also
Name	Indicates a name for the swimlane.	
Classifier	Indicates the classifier name for the swimlane; the (...) button allows specification of a classifier.	Select <Item> Dialog [880]
Back Color	Specifies a background color for a swimlane.	

5.5.6.21 *Swimlanes Matrix*

Enterprise Architect diagrams support a *Swimlanes Matrix* for all diagram types, based on the Zachman Framework.

The Swimlanes Matrix divides the diagram into cells of vertical columns and horizontal rows. The cell in the top left corner of the Swimlanes Matrix contains the heading of the matrix. The first cell at the top of each column contains the column title text. The first cell at the left of each row contains the row title text.

Access [Diagram | Swimlanes and Matrix > Matrix](#)

Use to

- Add a Matrix to a diagram

Reference

Field	Usage	See also
Active	Indicates that the Matrix dialog is active, as opposed to the Swimlanes dialog.	Swimlanes ^[75]
Title #	Indicates a title name, up to 3 title names can be specified.	
Color	Indicates a font color for a title.	
Font	Opens the font dialog, font styles can be specified here.	Set Element Font ^[83]
Hidden	Indicates if a title is hidden.	
Type	Indicates what type of matrix heading is created.	
Back Color	Indicates what background color the matrix item has.	
Hide Item	Indicates that the matrix item is hidden.	
Model Profiles	Indicates what profile is used during creation of the Matrix.	
Model Profiles : Save	Saves a profile to the Model Profiles list, prompting for a profile name.	
Model Profiles : Delete	Deletes a profile from the Model Profiles list.	
Lock	Indicates that the matrix item can not be moved or resized on a diagram.	
Line Widths	Indicates the width of the lines surrounding a matrix.	
Hand (Up)	If more than one item is present within a matrix, this button moves a highlighted element up one entry within the Matrix List.	
Hand (Down)	If more than one element is present within a matrix, this button moves a highlighted element down one	

	entry within the Matrix List.	
New	Allows for the creation of a new matrix item.	
Copy	Copies the highlighted matrix item.	
Delete	Deletes a matrix item from the Matrix List.	
Matrix List	A list of currently defined matrix items.	

Notes

- The heading is the first item in the list; you create only one heading
- When you define columns and rows, you define the header or title cells; the properties of these cells do not reflect on the matrix cells themselves - for example, the intersection cell of a column and row has a transparent background and therefore takes the color and shading effect of the diagram background
- You can also transport all the matrix profiles between models (as Diagram Matrix Profiles), using the Export Reference Data and Import Reference Data options
- By applying a Model Profile, you replace the current profile; save the current profile to avoid losing it
- To size the rows and columns, drag the row and column borders on the diagram - elements placed inside each cell are shifted when sizing; to prevent the elements shifting, press and hold (**Ctrl**) while sizing

Learn more

- [Export Reference Data](#) ^[306]
- [Import Reference Data](#) ^[308]

5.5.6.22 Using the Image Manager

The Image Manager dialog enables you to insert alternative images in diagrams, rather than inserting standard UML elements. For example, you might want to place a custom background image on a diagram, or display a custom image such as a Router or PC on a UML element.

Access **Right-click on an element within a diagram | Appearance | Select Alternate Image (Ctrl + Shift + W)**

To locate and display an image, click on individual image filenames, or press (**↑**) and (**↓**) to scroll through the list of images. As you highlight each image filename, the Preview panel changes to reflect the image. Double-click on the required image filename to display the image in full size.

Reference

On the Image Manager dialog, the following buttons are available:

Button	Usage	Shortcut	See also
View	Display the selected image in full size.	Alt + V	
Usage	Display the Element Usage [837] dialog, which lists the diagrams in which the selected image is used.		Element Usage [802]
Rename	Change the filename of the selected image. The option first warns you that the change would impact the other elements that use this image, and prompts you to confirm that you want to go ahead with the name change. If you continue, a prompt displays for you to enter the new filename.		Appearance Menu Section [837]
Add New	Browse appropriate directories to search for and import new images. You can import images in .BMP, .PNG, .EMF, .WMF, .TGA, .PCX or .JPG format. Internally, Enterprise Architect stores the images in .PNG or metafile format to conserve space. You can also add images by: <ul style="list-style-type: none"> • Importing a library of alternative images from the Sparx Systems website • Copying the Image Library from another model using the Export and Import Reference Data facilities, and • Capturing images from other sources into the clipboard and importing them into the Image Library (these are autonamed when they are added to the Image Library) If necessary, you can change the file name using the Rename option (above).	Alt + A	Import Image Library [758] Export Reference Data [308] Import Reference Data [308] Appearance Menu Section [837]
Update Selected	Refresh the selected image; for example, after it has been modified.	Alt + U	
Delete	(Not available for an image from a deployed MDG technology.) Delete the selected image. A message displays to indicate how many elements use the image. Click on the Continue button to delete information about the image from those elements, which then revert to their previous appearance.	Alt + D	

Button	Usage	Shortcut	See also
Cancel	Close the Image Manager dialog.		
OK	Confirm selection of the alternative image for the element selected in the diagram.	Alt + O	

Notes

- For elements with lifelines, such as those used on Sequence diagrams, the Lifeline must remain intact to enable messages to be created between the elements; therefore such elements cannot have alternative images
- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Configure Images** permission to configure alternative element images
- There are two categories of image available through the Image Manager: those images that have been imported into the model, which are stored in the model; and those images that are incorporated into a deployed MDG technology, which are not copied into the current model but are held in memory - the technology images are identified by the technology name followed by the image file name *MDGFrame::ovalshape.emf*
- If you are creating many elements of the same type that have a particular image, you should use a custom stereotype with an associated metafile
- You can transport image files between models, using the **Export Reference Data** and **Import Reference Data** options

Learn more

- [Create Custom Diagram Background](#)^[757]
- [Permission List](#)^[264]
- [Adding Images in MDG Technology](#)^[1333]
- [Custom Stereotype](#)^[1264]

5.5.6.22.1 Create Custom Diagram Background

Enterprise Architect diagrams have a single-color 'wash' background that you can set to a solid color or a fade gradient down the screen. You set the color on the Standard Colors page of the Options dialog, and whether to have a fade gradient on the Diagram Appearance page.

Alternatively, using the Image Manager dialog, you can create a non-tiled background for diagrams.

How to

To create a non-tiled background follow the steps below:

Step	Action	See also
1	Create a Boundary object from the Use Case Elements page of the Toolbox; do not use the Boundary element from any other section of the Toolbox.	System Boundary ^[1170] Toolbox ^[698]
2	Stretch the Boundary to a size that can contain all of the elements you intend to place on the diagram, and drag it to the edges of the diagram workspace.	
3	Right-click on the Boundary element and select the Z-Order Send to Bottom context menu option. This ensures that the Boundary is not displayed in front of any other element in the diagram.	
4	Either: <ul style="list-style-type: none"> • Press (Ctrl+Shift+W), or • Right-click on the Boundary to display the context menu, and select the Appearance Alternate Image menu option 	
5	On the Image Manager dialog, select an appropriate image as the diagram background and ensure that the image size is large enough to span the required size of the diagram background.	Image Manager ^[755]
6	When you have selected the required image, click on the OK button. Alternatively, you can copy an image from another source onto the Windows clipboard, right-click on the Boundary element in the Enterprise Architect diagram, and select the Appearance Apply Image From Clipboard context menu option.	

Learn more

- [Standard Colors](#)^[540]
- [Diagram Appearance](#)^[550]

5.5.6.22.2 Import Image Library

Using the Image Library enables you to create attractive diagrams with custom images. A bundled clip art collection of UML-based images is available as an Imported Image Library, from the Sparx Systems website. Image libraries enable you to import a collection of images into the Image Manager in one process.

How to

To import the Image Library, follow the steps below (before starting, you must have a suitable Image Library file):

Step	Action	See also
1	Download the Image Library from the Sparx Systems website.	www.sparxsystems.com/resources/image_library.html
2	Select the Project Model Import/Export Import Reference Data menu option. The Import Reference Data dialog displays.	
3	Locate the XML Image Library file to import using the Select File button. The file name is <i>ImageLibrary.xml</i> in the directory in which you saved the file.	
4	Select the data set containing the Image Library. Then click on the Import button.	

To use the images contained within the Image Library

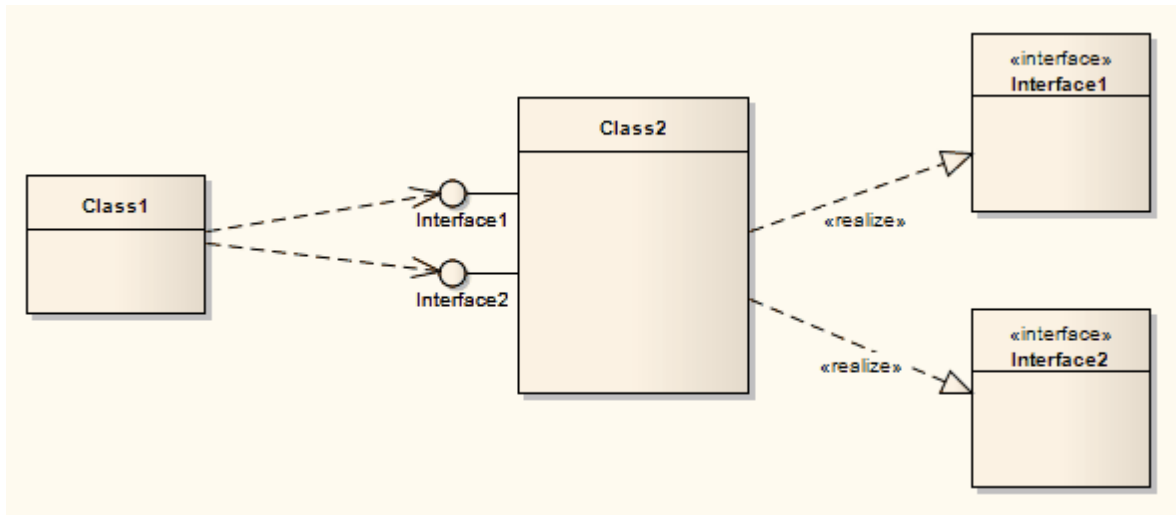
Step	Action	See also
1	Create a diagram to associate with the images contained in the Image Library.	
2	Select the element to change from the default appearance to one of the images contained within the library.	
3	Either: <ul style="list-style-type: none"> • Press (Ctrl+Shift+W), or • Right-click on the selected element and select the Appearance Select Alternate Image context menu option 	
4	On the Image Manager dialog, in the Name field highlight the appropriate image name and click on the OK button.	

Notes

- Images contained within the Image Library are copyright of Sparx Systems, are only available for use in conjunction with Enterprise Architect, and are supplied on the understanding that they are not used under any other circumstance

5.5.6.23 Show Realized Interfaces of Class

You can display each interface directly realized by a Class as a 'lollipop' style interface node, which protrudes from the left-hand side of the Class. Connectors can be directly attached to the node, indicating usage of the interface part of the Class or component. See the example below:



Topics

Topic	Detail	See also
Explanation	<p>In this example, <i>Class2</i> realizes <i>Interface1</i> and <i>Interface2</i> as represented by the interface nodes protruding from the Class.</p> <p><i>Class1</i> is dependent on these two interfaces, which is shown by the Dependency connectors linking to the nodes.</p> <p>To show nodes for the interfaces a Class realizes, as in the above diagram, right-click on the Class and select the Embedded Elements Show Realized Interfaces context menu option.</p> <p>This setting only applies to the selected Class, and can be changed at any time.</p>	

5.5.6.24 Label Menu Section

You can add labels to both connectors and elements, using the element or connector context menu as follows:

- Element:
 - Select the **Embedded Elements** menu option and either the **Add <element>** option or the **Embedded Elements** option; the label is the embedded element name
 - Apply an alternative image to an element (that might also have a run state; the run state, attributes and operations of the element are then displayed as a label of the element)
- Connector - Select the **Properties** option and define the connector name, stereotype, constraints and/

or source and target roles

Once you have these labels, you can edit and format them using the **Labels** context menu.

Access **Right-click on a label**

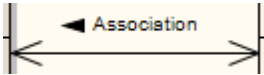
Reference

The **Labels** context menu associated with embedded elements provides the following options:

Action	Usage	Shortcut	See also
Set Label Color	Specify a color for the label.		
Hide Label	Hide the label; to unhide the label, right-click on the element and select the Appearance Show labels context menu option.		Appearance Menu section ^[837]
Bold	Set the label font to bold.		
Text Alignment	Align the text within the label text area. The options available from the submenu enable you to specify left, center and right alignment.		
Label Rotation	Orient the label in the horizontal or vertical planes, with the vertical plane offering the option of clockwise or anti-clockwise position.		
Default Position	Move the label to the initial default location.		
Default Color	Set the label color to the default color.		

The **Labels** menu associated with connectors provides the following options:

Action	Usage	Shortcut	See also
Set Label Color	Specify a color for the label.		
Hide Label	Hide the label; to unhide the label use the Visibility Set Label Visibility ^[957] option on the connector context menu.		Set Label Visibility ^[957]

Action	Usage	Shortcut	See also
Bold	Set the label font to bold.		
Text Alignment	Align the text within the label text area. The options available from the submenu enable you to specify left, center and right alignment.		
Label Rotation	Orientate the label horizontally or vertically and, if vertically, in a clockwise or anti-clockwise position.		
Direction	Set a small arrow at the end of the label pointing to either the label source or the destination dependent upon selection from the available options.  This is part of the label, so if there is no label there is no direction indicator.		
Default Position	Move the label to the default location.		
Default Color	Set the label color to the initial default color.		

Notes



- As labels can be concentrated on and around the element or connector, make sure that you click on a section of the required label that is clear of any other label or structure

Learn more

- [Embedded Elements](#) ^[834]
- [Using the Image Manager](#) ^[755]
- [Run-time State](#) ^[1198]
- [Connector Properties](#) ^[961]

5.5.6.25 Pan and Zoom a Diagram

Topics

Topic	Detail	See also
Pan	<p>Pan across the Diagram View in the following ways:</p> <ul style="list-style-type: none"> • Use (←), (→), (↑), (↓), (Page Up), (Page Down), (Home) and (End) when the Diagram View is selected • Use the scrollbars • Use the middle mouse button • Use the Pan & Zoom window 	Pan & Zoom Window ^[610]
Zoom	<p>You can zoom into and out from a diagram using the zoom buttons on the diagram toolbar, or by using the Diagram Zoom submenu.</p>  <p>Change the zoom level by 10% by clicking on either the Zoom In (+) or Zoom Out (-) buttons.</p> <p>Alternatively, select the Zoom In or Zoom Out options from the Diagram Zoom submenu.</p> <p>There are three ways to return the diagram to 100%:</p> <ul style="list-style-type: none"> • Click on the  button • Select Zoom to 100% from the Diagram Zoom submenu • (Ctrl) + middle-click the mouse 	

Notes

- You can zoom in and out of the main window dynamically by holding (**Ctrl**) and rolling the mouse wheel
- Changes in diagram magnification through the zoom options can be saved as permanent changes to the diagram
- At high levels of zoom, element features cease to display. This is because of the difficulty the Windows font mapper has in choosing a font for extreme conditions, and the result can look odd

5.5.6.26 Move Elements In Diagram Sections

As you build up a diagram, you might find that you have to move part of the diagram up, down or to one side. You can do this in one of two ways:


Topics

Topic	Detail	See also
Moving Elements	<ul style="list-style-type: none"> Hold the left mouse button down and drag over a group of elements to move (creating an outline around the elements), then click on an element in the outline and move the group as required Press (Alt) and click on the diagram, then drag the cursor to move everything beyond the cursor in the direction of the movement. <p>The first method enables you to reposition groups of elements within the larger diagram. The second method enables you to create space within the diagram without pushing some elements into others, as might happen if you cannot see the whole diagram on one screen.</p> <p>When you press (Alt) and click on the diagram, as you move the cursor a line displays on the diagram just behind the cursor. If you are moving the cursor left, everything to the left of the line moves with the cursor. If you move the cursor up, everything above the line moves up.</p> <p>However, if you move the cursor diagonally, two lines display to create a quadrant, and everything within the quadrant moves. For example, if you move the cursor left and down, everything below and left of the cursor moves.</p>	
Fine Movement	To adjust (or 'nudge') the position of a single element or a selected group of elements, press (Shift) + (→), (←), (↑) or (↓).	

5.5.6.27 View Last and Next Diagram

Enterprise Architect enables you to step backwards and forwards through the currently-open diagrams, including the Start Page.

Topics

Topic	Detail	See also
Usage	 <p>To view the previous or next diagram use the Previous or Next buttons on the Diagram toolbar.</p> <p>Use the Home button to display the default project diagram (if one has been specified).</p>	Default Project Diagram ^[744]

5.5.6.28 Set Diagram Page Size

You can change the size of the diagram area (or scrollable/printable area) using the Diagram Properties dialog.

How to

To set the page size

Step	Action	See also
1	Load a diagram.	
2	Double-click on the background to open the Diagram Properties dialog.	
3	Click on the Diagram tab and, in the Appearance panel ensure that the Show Page Border checkbox is selected.	
4	On the Page Setup panel, click on the Advanced button. The Print Advanced dialog displays.	
5	Click on the Page Setup button. The Page Setup dialog displays. As you adjust the settings on this dialog, the page icon at the top illustrates the effects of your changes.	
6	In the Size field, click on the drop-down arrow and select an appropriate page size.	
7	In the Orientation panel click on the radio button for the orientation of the page to print.	
8	In the Margins panel, type the required left, right, top and bottom page margins for the diagram, in inches.	
9	Click on the OK button on the Page Setup dialog, the Print Advanced dialog, and the Diagram Properties dialog, in turn. The area within the page boundary lines on your diagram is expanded or reduced accordingly. When you print or print preview, the output is cropped to these boundary lines and the diagram divided between the necessary number of pages.	

Topics

Topic	Detail	See also
Setting the Default Paper Size for New Diagrams	<p>You can set the default paper size for new diagrams on the Diagram page of the Options dialog (select the Tools Options Diagram menu option).</p> <p>Once the paper size is set there, all new diagrams have that as the default size.</p>	Configure Local Options - Diagram ^[549]

Topic	Detail	See also

Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Update Diagrams** permission to change diagram page setup

Learn more

- [Permissions List](#)^[264]

5.5.6.29 Scale Image to Page Size

When you print a diagram, the default setting is to scale the image to fit the size of the printer paper you have defined in the page set-up. The image is not scaled up to fill the page, but it is scaled down if it exceeds the current page boundary. The image retains its current proportions; that is, it is scaled down equally in the X and Y dimensions. For a large diagram, this can mean that the components of the diagram are small and hard to read.

Alternatively, you can print a multi-page image; that is:

- allow the diagram image to print on as many printer pages as it naturally occupies, (no scaling), or
- scale the diagram image to exactly fit a specified number of pages

In all three cases you also define the paper size and orientation.

How to

To turn off or customize image scaling options

Step	Action	See also
1	Select the diagram to scale.	
2	Either: <ul style="list-style-type: none"> • Double-click on the diagram background, or • Right-click on the background and select the Properties context menu option <p>The <type> Diagram: <name> dialog displays.</p>	
3	Click on the Diagram tab and, in the Page Setup panel click on the Advanced button.	Page size and Alignment ^[764]

Step	Action	See also
	The Print Advanced dialog displays.	
4	<p>Select from the following options as required:</p> <ul style="list-style-type: none"> • None: select to print on as many pages as the diagram image covers • Scale to 1 page: select to scale the diagram image to fit on the currently selected page • Custom: select to specify the width and height of the diagram images across a specified number of pages • Page Setup: click to select the page size and alignment 	

Notes

- Before printing, make sure you have selected the required page layout using the **Page Setup** button

Learn more

- [File Menu](#) ⁹²¹

5.5.6.30 Lock Diagram

You can lock a diagram against inadvertent changes, such as moving or sizing elements.

How to

To lock a diagram

Step	Action	See also
1	Open the diagram to lock.	
2	Right-click on the background to open the diagram context menu.	
3	Click on the Lock Diagram option to prevent further changes.	
4	<p>Click on the OK button.</p> <p>If a user selects an item on a locked diagram, the object border or outline displays in red.</p>	

Notes

- This does not apply in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions if security is enabled; in that case, see the *Lock Model Elements* topic


Learn more

- [Lock Model Elements](#) ^[27]

5.5.6.31 Undo Last Action

When editing diagrams, Enterprise Architect supports multiple undo levels for moving, re-sizing and deleting elements, and for deleting connectors.

Topics

Topic	Detail	See also
Usage	There are three ways to undo the last action: <ul style="list-style-type: none"> • Press (Ctrl+Z) • Select the Edit Undo menu option • Click on the Undo button in the Default Tools toolbar -  	


Notes

- Currently you cannot undo element additions or connector moves

5.5.6.32 Redo Last Action

When editing diagrams, Enterprise Architect supports multiple undo levels for moving, re-sizing and deleting elements, and for deleting connectors. If an Undo action is in error, you can redo the action to reverse the Undo.

Topics


Topic	Detail	See also
Usage	There are three ways to redo the last action: <ul style="list-style-type: none"> • Press (Ctrl+Y) • Select the Edit Redo menu option • Click on the Redo button in the Default Tools toolbar -  	


5.5.7 Layout Diagrams

Enterprise Architect provides a Layout Tools window to enable you to lay out the elements in a diagram. Unless the diagram is very simple, this facility does not lay out the entire diagram; it consists of a set of tools to set out different areas or sets of elements in the diagram.

How to

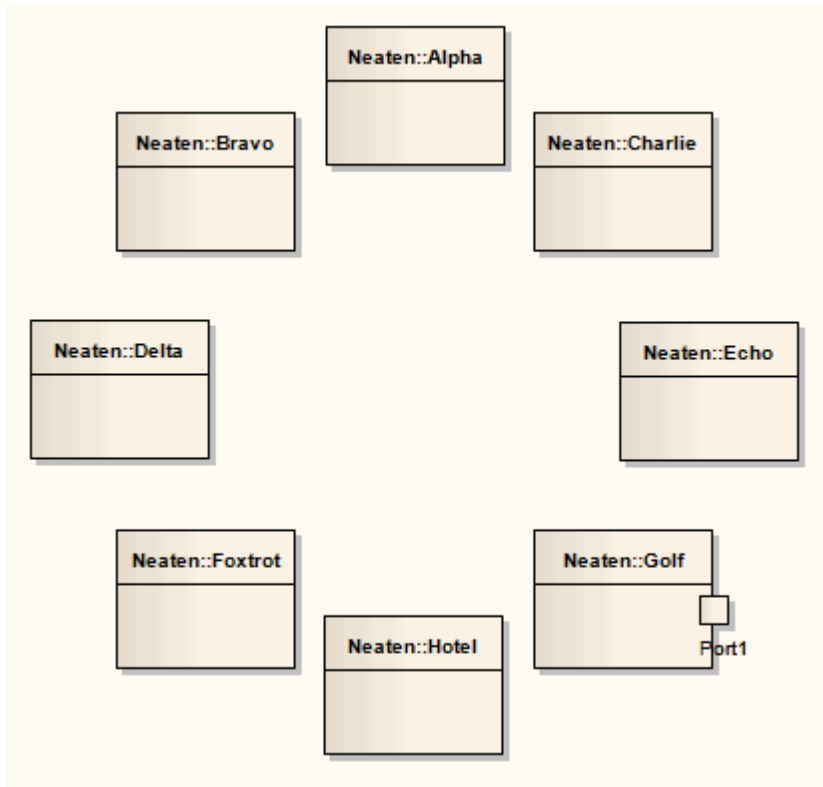
To start to lay out a diagram

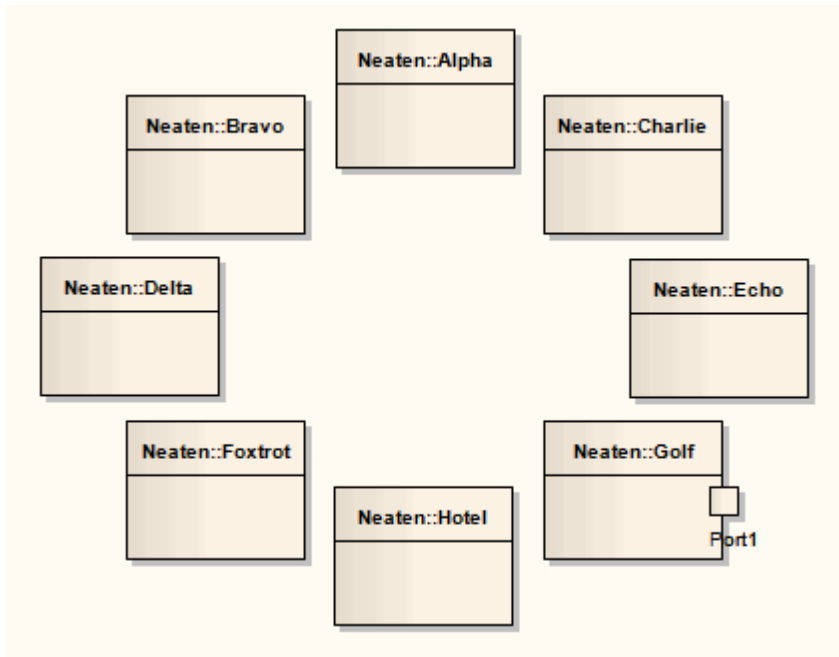
Step	Action	See also
1	Open the Layout Tools window; either: <ul style="list-style-type: none"> • Select the View Diagram Layout menu option, or • Right-click on the main toolbar and select the Layout Tools context menu option 	
2	Select the elements to lay out on the currently-active diagram - hold (Shift) or (Control) while you click on each required element, or hold the mouse button down while you sweep over the area containing the required elements. If no elements are selected on the active diagram, then all elements on the diagram are laid out (except where otherwise documented).	
3	Click on the drop-down arrow on the top left field of the Layout Tools window, and select the required layout type.	
4	The layout type determines the fields presented in the window, therefore the appropriate fields are described in the topic for each layout type <ul style="list-style-type: none"> • Circle/Ellipse • Box • Per Page • Digraph • Spring • Neaten • Converge/Diverge • Fan Relations • Auto Route 	Circle/Ellipse ^[770] Box ^[770] Per Page ^[770] Digraph ^[770] Spring ^[780] Neaten ^[781] Converge/Diverge ^[782] Fan Relations ^[783] Auto Route ^[780]
5	When you have completed the fields, click on the  button. Enterprise Architect sets out the selected elements according to the options you have chosen.	
6	If you do not want to work with the new layout, click on the Undo button in	Automatically Laying

Step	Action	See also
	<p>the toolbar ().</p> <p>Enterprise Architect also provides a facility for automatically laying out a diagram; if necessary, you can manually adjust the final result of this automatic process.</p>	<p>out a Diagram ⁷⁸⁷</p>

5.5.7.1 Circular/Elliptical Layout

The *Circle* and *Ellipse* layouts arrange the selected elements in a circle or elliptical pattern, using the largest horizontal and vertical element edge in the set of elements when calculating the radius of the layout arc.





How to

To invoke the Circular or Ellipse layout

Step	Action	See also
1	Follow the general <i>Layout Diagrams</i> procedure, and at step 3 select either Circle or Ellipse as required.	Layout Diagrams 769
2	<p>Click on the drop-down arrow in the Sort By: field and select the required sort parameter:</p> <ul style="list-style-type: none"> • None - Elements are passed to the specified layout in the order in which they appear on the original diagram (left to right, top to bottom) • Area (Ascending) - Elements are passed to the specified layout in order of the screen space they occupy, smallest to largest • Area (Descending) - Elements are passed to the specified layout in order of the screen space they occupy, largest to smallest • Name (Ascending) - Elements are passed to the specified layout in alphanumeric order, based on the element name • Name (Descending) - Elements are passed to the specified layout in reverse alphanumeric order, based on the element name • Element Type - Elements are grouped by type (for example, Class, Use Case) and in alphanumeric order within the group by name 	

Step	Action	See also
3	Under the Placement option, select either: <ul style="list-style-type: none">• Top to Bottom (the elements are positioned in the required order, zig-zagged across the perimeter of the circle or ellipse; see Diagram A, below)• Circular (the elements are placed in the required order, clockwise around the perimeter of the circle or ellipse; see Diagram B, below)	
4	Select the Center focused element checkbox to put the last-selected element (the one with the hashed border) in the center of the circle or ellipse; see Diagram C, below.	

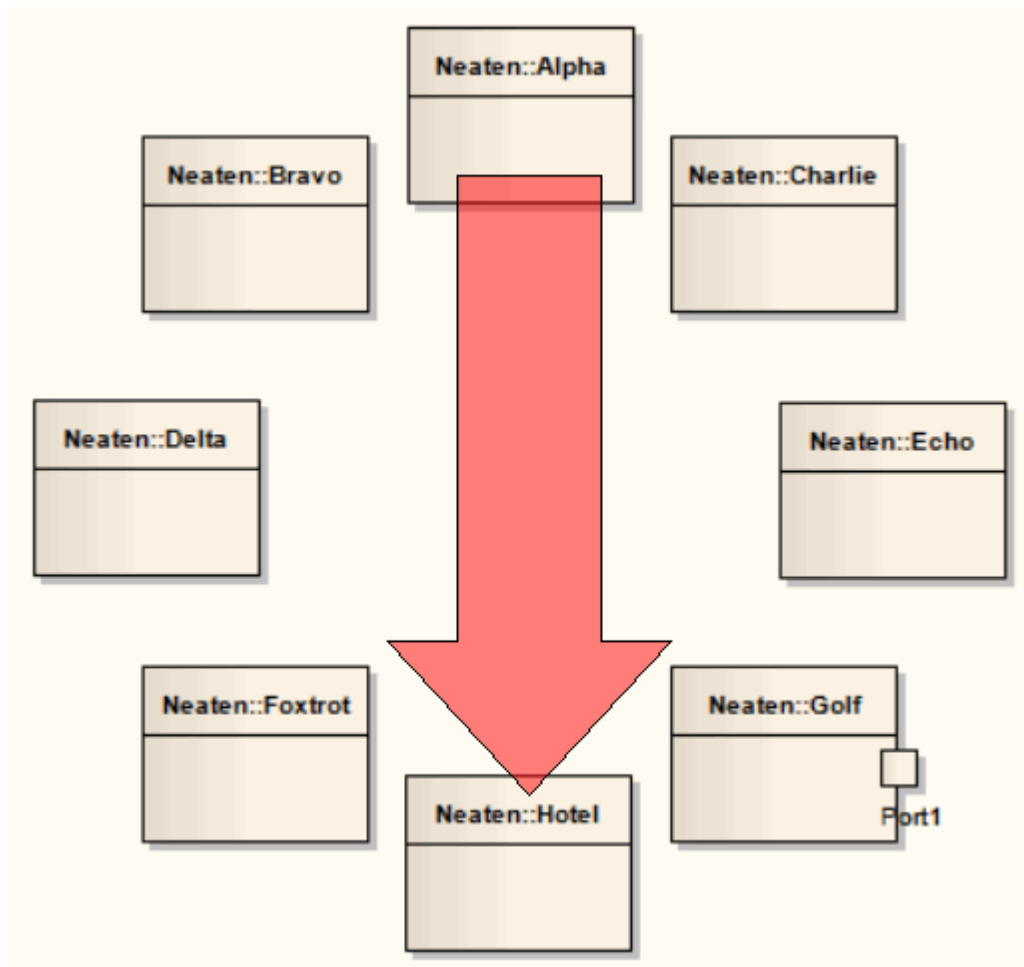
Diagram A - Top To Bottom Layout

Diagram B - Circular Layout

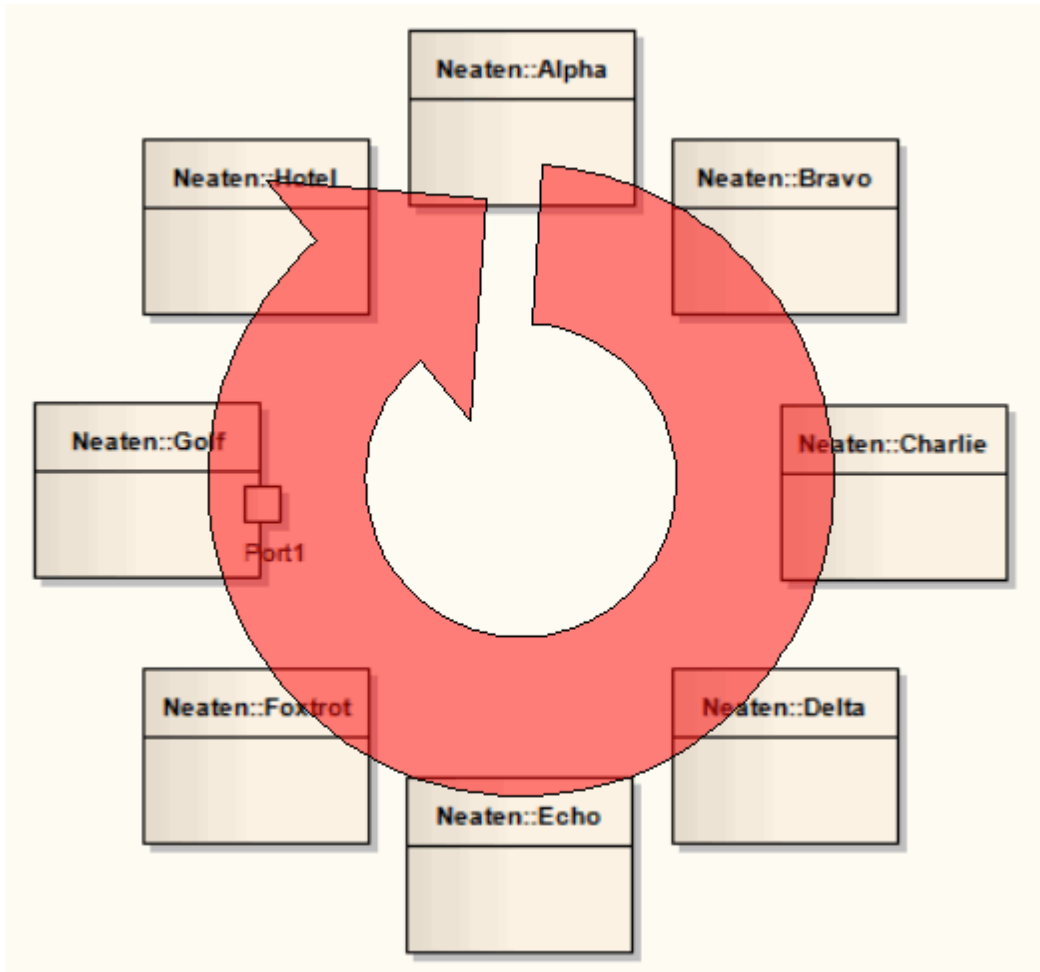
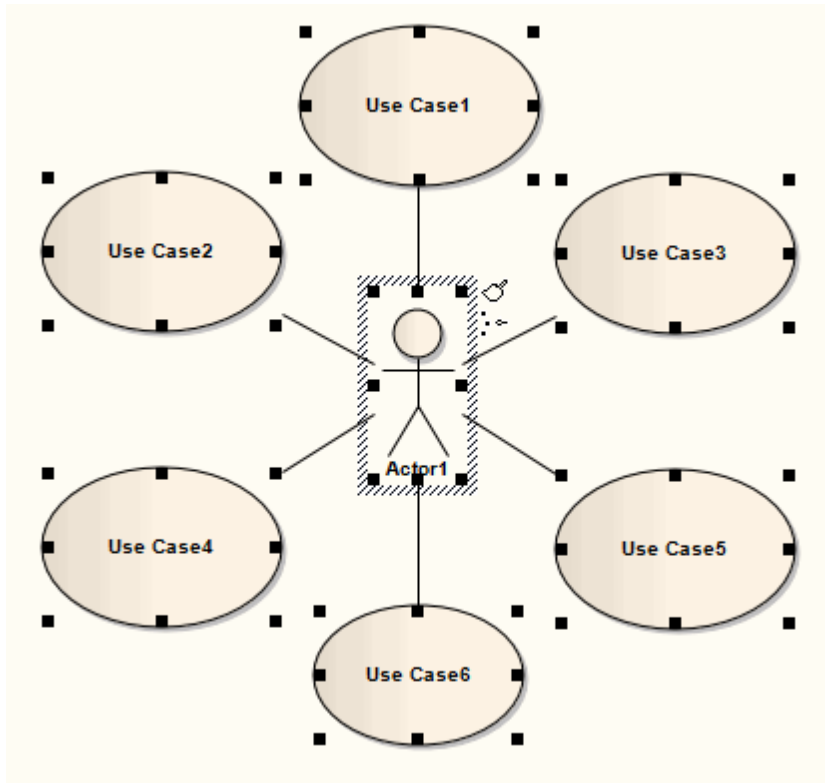
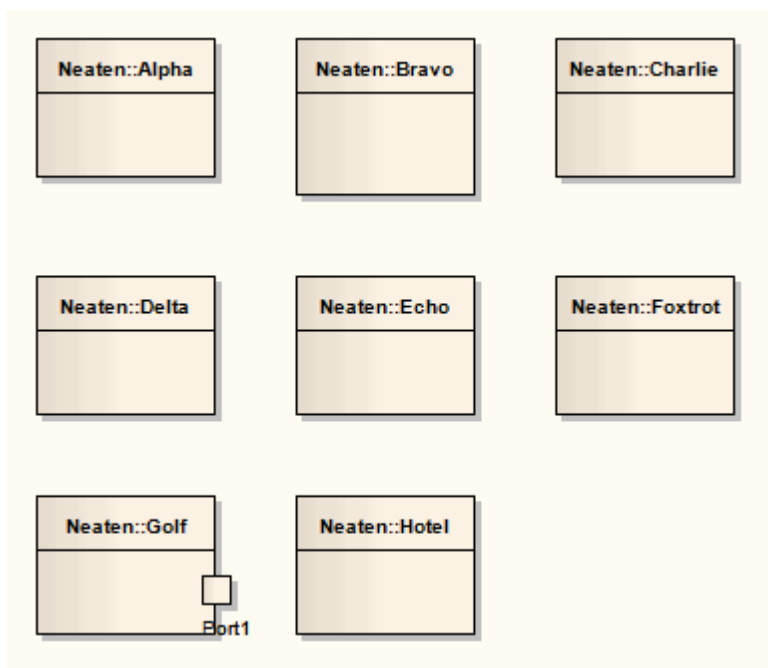


Diagram C - Center Focused Element

5.5.7.2 Box Layout

The *Box* layout arranges the set of selected elements into a square grid.



The dimensions of the box are determined by the square root of the number of selected elements (for example, 16 elements create a 4x4 box).

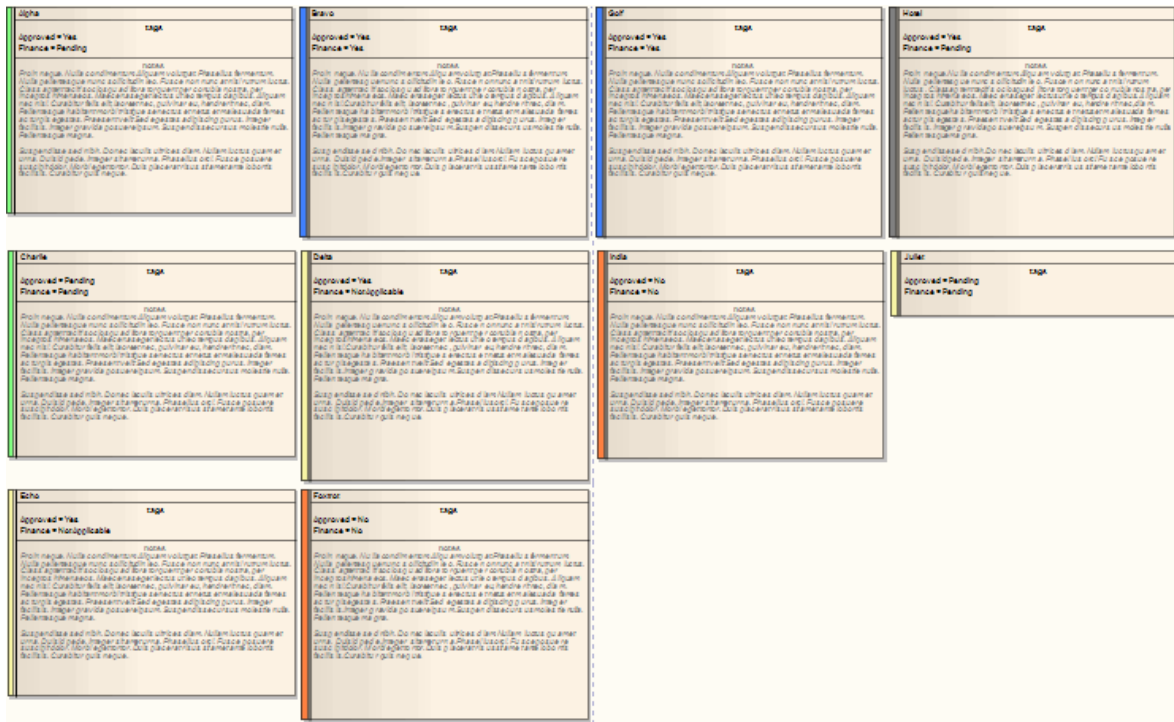
How to

To invoke the Box layout

Step	Action	See also
1	Follow the general <i>Layout Diagrams</i> procedure, and at step 3 select Box .	Layout Diagrams 769
2	Click on the drop-down arrow in the Sort By: field and select the required sort parameter: <ul style="list-style-type: none"> • None - Elements are passed to the specified layout in the order in which they appear on the original diagram (left to right, top to bottom) • Area (Ascending) - Elements are passed to the specified layout in order of the screen space they occupy, smallest to largest • Area (Descending) - Elements are passed to the specified layout in order of the screen space they occupy, largest to smallest • Name (Ascending) - Elements are passed to the specified layout in alphanumeric order, based on the element name • Name (Descending) - Elements are passed to the specified layout in reverse alphanumeric order, based on the element name • Element Type - Elements are grouped by type (for example, Class, Use Case) and in alphanumeric order within the group by name 	
3	In the Padding (px) field, type the vertical and horizontal distance between elements, in pixels.	
4	Select the appropriate element distribution option: <ul style="list-style-type: none"> • Automatically distribute: Automatically calculate the dimensions of the box (the square root of the number of selected elements; for example, 16 elements create a 4x4 box) • Specify distribution: Manually define the width of the box, in columns 	
5	If you selected Specify Distribution , in the Columns field type the required number of columns.	

5.5.7.3 Per Page Layout

The *Per Page* layout divides each diagram page into a number of cells, which house the selected elements. The number of cells per page is determined by the page distribution parameter, as explained below.



How to

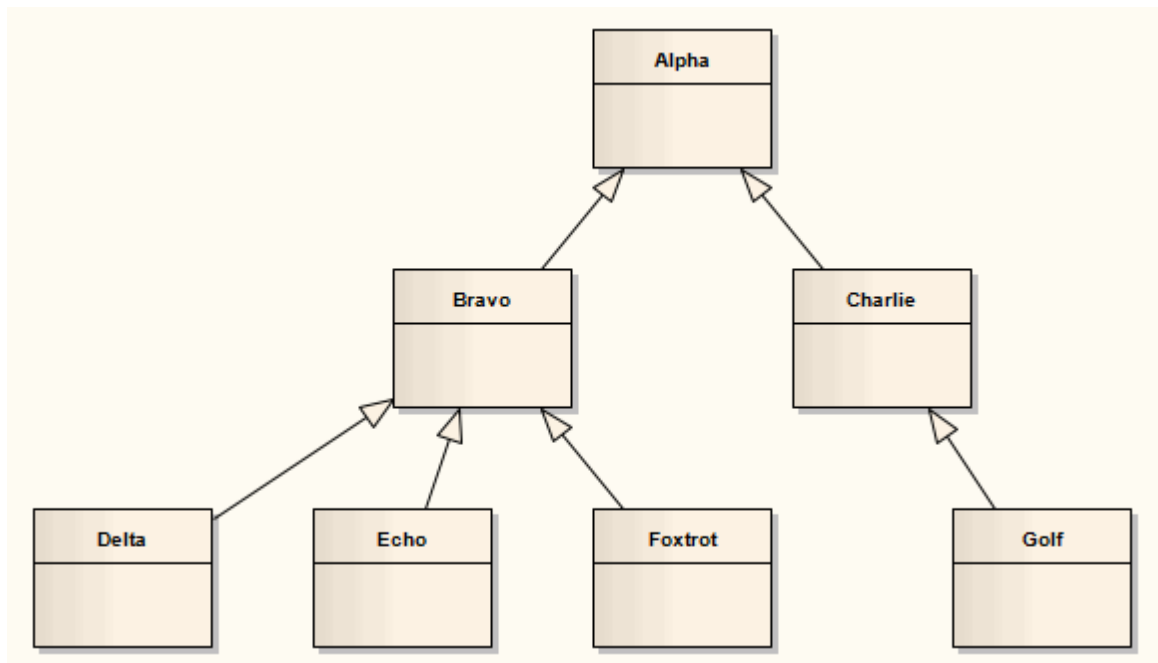
To invoke the Per Page layout

Step	Action	See also
1	Follow the general <i>Layout Diagrams</i> procedure, and at step 3 select Per Page .	Layout Diagrams ⁷⁶⁹
2	Click on the drop-down arrow in the Sort By: field and select the required sort parameter: <ul style="list-style-type: none"> • None - Elements are passed to the specified layout in the order in which they appear on the original diagram (left to right, top to bottom) • Area (Ascending) - Elements are passed to the specified layout in order of the screen space they occupy, smallest to largest • Area (Descending) - Elements are passed to the specified layout in order of the screen space they occupy, largest to smallest • Name (Ascending) - Elements are passed to the specified layout in alphanumeric order, based on the element name • Name (Descending) - Elements are passed to the specified layout in reverse alphanumeric order, based on the element name • Element Type - Elements are grouped by type (for example, Class, Use Case) and in alphanumeric order within the group by name 	

Step	Action	See also
3	In the Padding (px) field, type the vertical and horizontal distance between cells, in pixels.	
4	Select the appropriate page distribution option: <ul style="list-style-type: none">• Automatically distribute: Automatically calculate the optimum number of cells, taking into consideration the largest horizontal and vertical element edges• Specify distribution: Manually enter the per page grid dimensions	
5	If you selected Specify Distribution , in the Rows and Columns fields type the required number of rows and columns.	
6	Select the Center Elements checkbox to place each element in the center of its cell; otherwise the element placement defaults to the top left corner of the cell.	
7	In the Start Page field, type the number from which to start page numbering; pages begin at the top left and continue horizontally to the right.	

5.5.7.4 Digraph Layout

The *Digraph* layout arranges the selected elements into a directed graph (digraph for short). The Digraph attempts to highlight the hierarchy of the elements while keeping the direction of all connectors pointing to the same edge of the diagram.



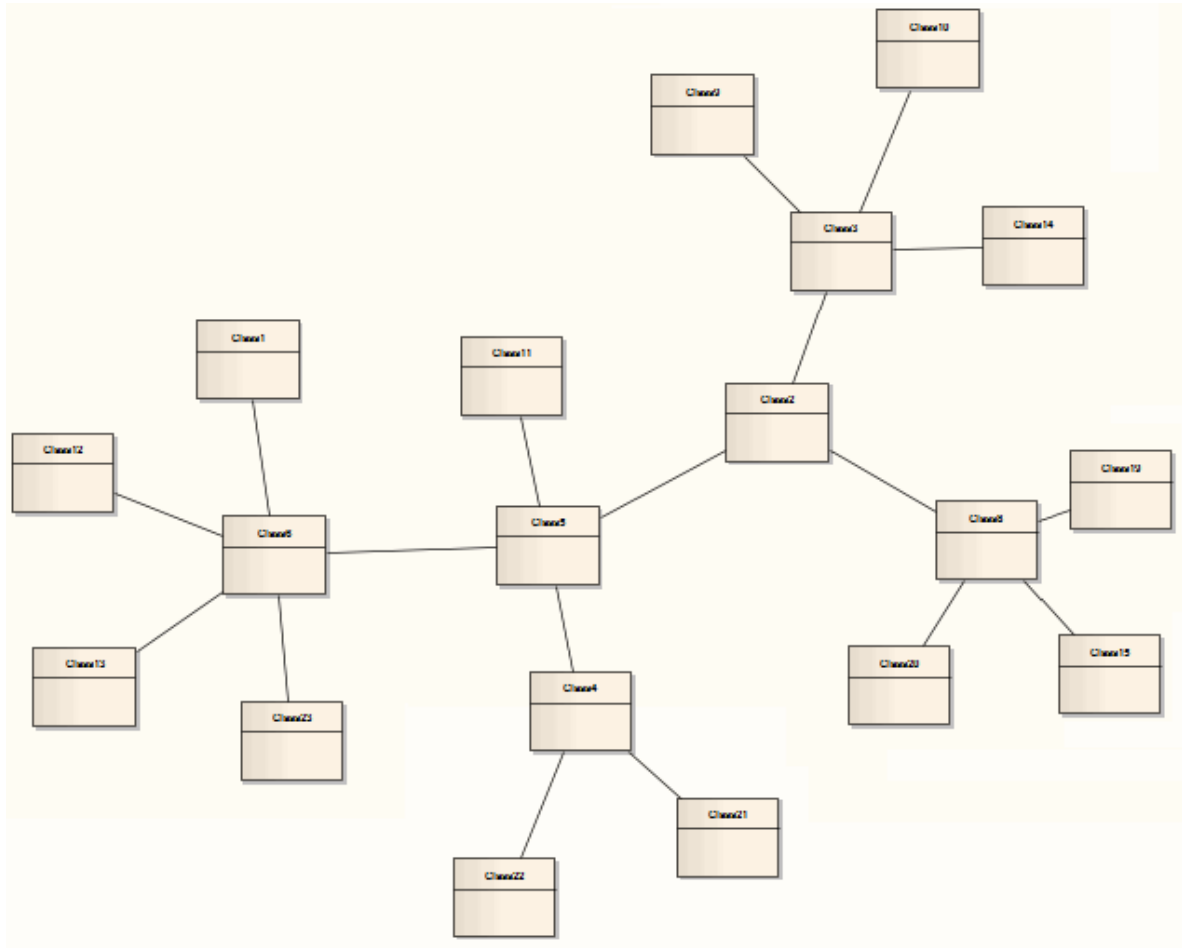
The Digraph layout provides the same behaviour as the Automatic Diagram layout.

Learn more

- [Layout Diagrams](#)^[769]
- [Layout a Diagram Automatically](#)^[787]

5.5.7.5 Spring Layout

The *Spring* layout uses a force-directed approach to arrange the selected elements organically.



The Spring layout employs a physical analogy to lay out elements. Each element is treated as a particle with a like electrical charge that repels other elements. Connectors act as springs (hence the term Spring layout) that draw connected elements back together. The layout is good for highlighting clusters of related objects and identifying symmetry in the graph.

How to

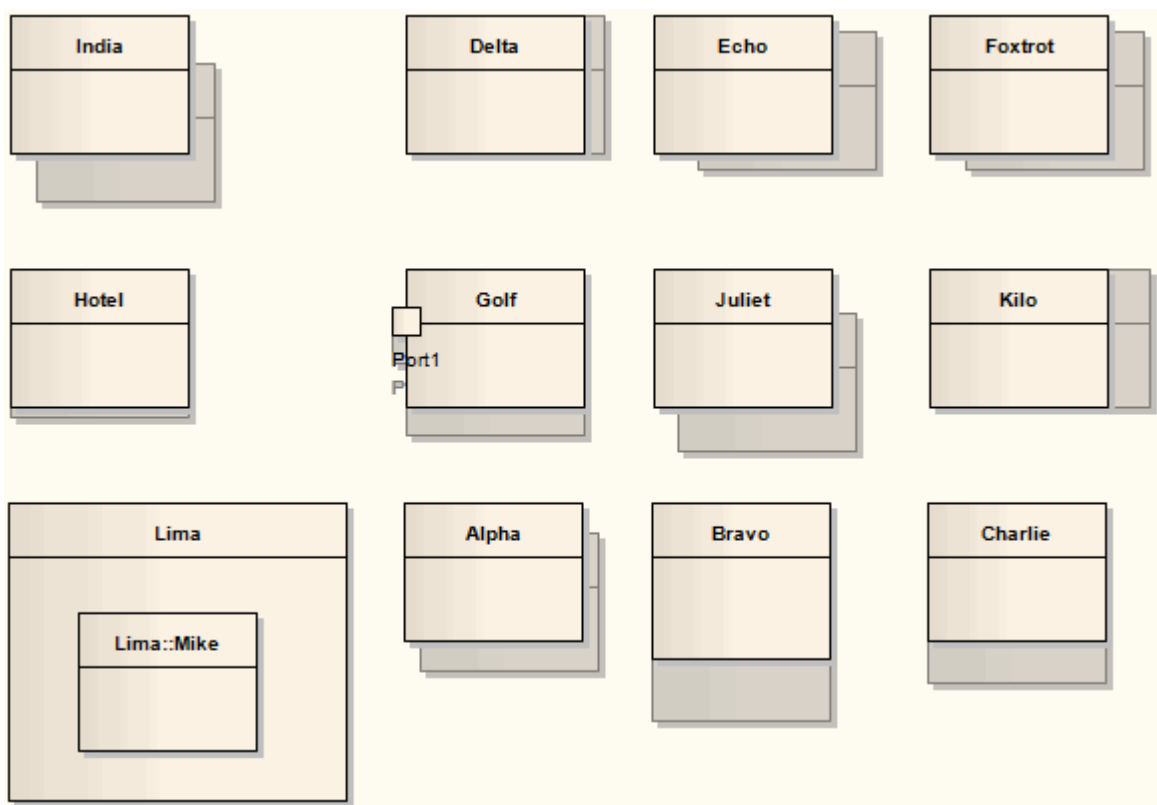
To invoke the Spring layout

Step	Action	See also
1	Follow the general <i>Layout Diagrams</i> procedure, and at step 3 select Spring .	Layout Diagrams ¹⁷⁶⁹
2	In the Iterations field, type the number of iterations, or rounds, to perform to reach	

Step	Action	See also
	<p>the final layout.</p> <p>The layout is developed over several iterations; depending on the complexity of the graph, increasing the number of iterations produces a better result but takes longer to calculate.</p>	
3	<p>If the diagram contains elements that significantly vary in size, and that might overlap in the final layout, select the Scale to prevent overlap checkbox to scale up the positions of the selected elements (preserving size) until no elements overlap.</p>	

5.5.7.6 Neaten Layout


The *Neaten* layout attempts to arrange the selected elements into a grid based on their horizontal and vertical proximity to each other. Elements that share the same row or column are aligned based on the **Column** and **Row Snap** parameters.



How to

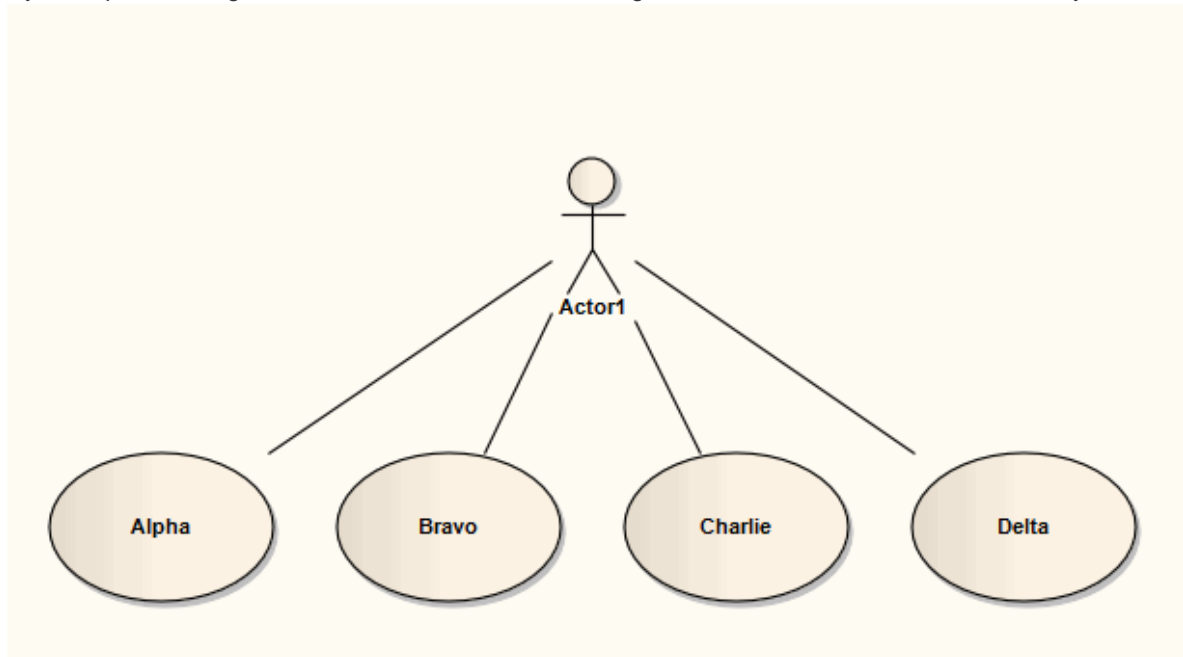
To invoke the Neaten layout

Step	Action	See also
1	<p>Follow the general <i>Layout Diagrams</i> procedure, and at step 3 select Neaten.</p>	Layout

Step	Action	See also
		Diagrams 
2	<p>In the Threshold (px) field, type the height or width distance, in pixels, at which elements should be considered to be in the same row or column.</p> <p>A lower threshold value produces a tighter result, with only elements that are extremely similar - vertically or horizontally - considered to be in the same row or column.</p>	
3	<p>In the Column Snap field, click on the drop-down arrow and select the appropriate alignment for elements in the same column:</p> <ul style="list-style-type: none">• Left - elements are aligned with the left edge of the left-most element in the column• Center - elements are aligned with the vertical center of the center-most element in the column• Right - elements are aligned with the right edge of the right-most element in the column	
4	<p>In the Row Snap field, click on the drop-down arrow and select the appropriate alignment for elements in the same row:</p> <ul style="list-style-type: none">• Top - elements are aligned with the top edge of the highest element in the row• Center - elements are aligned with the horizontal center of the center-most element in the row• Bottom - elements are aligned with the bottom edge of the lowest element in the column	

5.5.7.7 Fan Relations Layout

The *Fan Relations* layout arranges the immediate relations of an element around a specified edge. This layout requires a single element to be selected on the diagram, to be used as the context for the layout.



How to

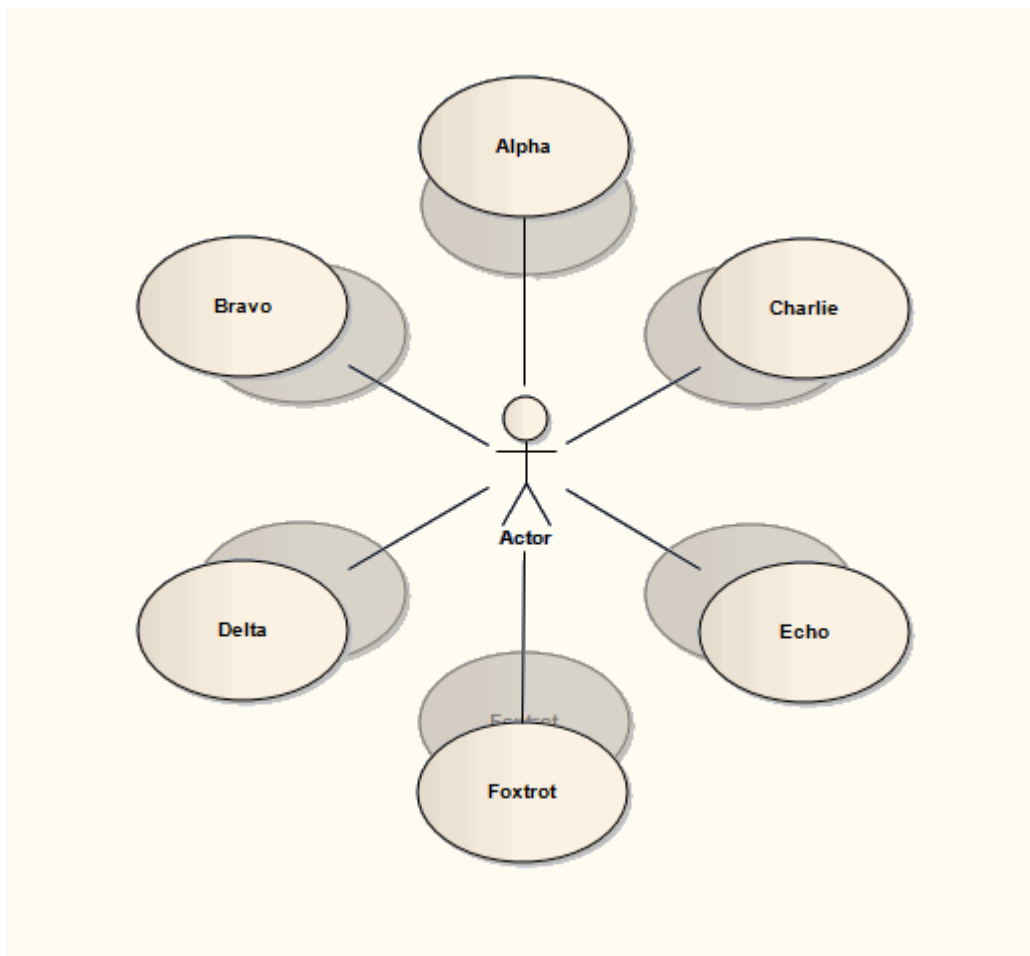
To invoke the Fan Relations layout

Step	Action	See also
1	Open the Layout Tools window; either: <ul style="list-style-type: none"> • Select the View Diagram Layout menu option, or • Right-click on the main toolbar and select the Layout Tools context menu option 	Layout Diagrams 769
2	Select the single element around which to lay out related elements on the currently-active diagram.	
3	Click on the drop-down arrow on the top left button of the Layout Tools window, and select Fan Relations .	
4	Click on the drop-down arrow in the Sort By: field and select the required sort parameter. The options are: <ul style="list-style-type: none"> • None - Elements are passed to the specified layout in the order in which they appear on the original diagram (left to right, top to bottom) 	

Step	Action	See also
	<ul style="list-style-type: none"> • Area (Ascending) - Elements are passed to the specified layout in order of the screen space they occupy, smallest to largest • Area (Descending) - Elements are passed to the specified layout in order of the screen space they occupy, largest to smallest • Name (Ascending) - Elements are passed to the specified layout in alphanumeric order, based on the element name • Name (Descending) - Elements are passed to the specified layout in reverse alphanumeric order, based on the element name • Element Type - Elements are grouped by type (for example, Class, Use Case) and in alphanumeric order within the group by name 	
5	In the Padding (px) field, type the separation required between the selected element and its related elements, in pixels.	
6	Select the connector direction to use in determining the related elements to lay out; either: <ul style="list-style-type: none"> • Incoming Nodes - to lay out related elements that have the selected element as the target • Outgoing Nodes - to lay out related elements that have the selected element as the source 	
7	In the Fan Edge field, click on the drop-down arrow and specify the edge of the selected element from which to lay out the related elements: <ul style="list-style-type: none"> • Left - to arrange related elements to the left of the selected element • Right - to arrange related elements to the right of the selected element • Top - to arrange related elements from the top of the selected element • Bottom - to arrange related elements from the bottom of the selected element 	


5.5.7.7.1 Converge/Diverge Layout

The *Converge* layout attracts the set of selected elements towards the center of their bounding rectangle. Conversely, the *Diverge* layout repels the set of selected elements away from the center of their bounding rectangle. The Converge/Diverge layout also tries to maintain connector angles if an element in the set contains a connector with waypoints.



How to

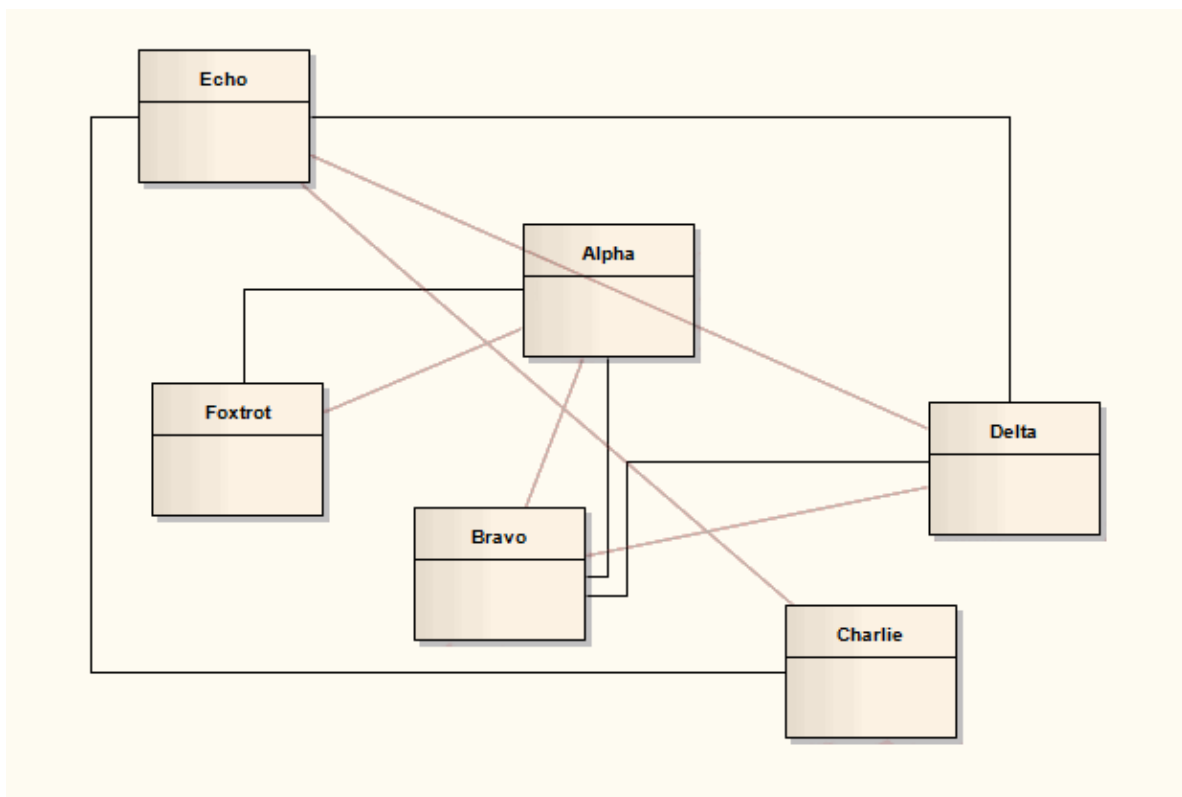
To invoke the Converge/Diverge layout

Step	Action	See also
1	Follow the general <i>Layout Diagrams</i> procedure, and at step 3 select Converge/Diverge .	Layout Diagrams 
2	For Direction , select the required layout: <ul style="list-style-type: none"> • Converge - attracts the set of selected elements to the center point • Diverge - repels the set of selected elements from the center point 	
3	The Amount (%) slider determines how far the elements are moved towards or away from the center point; the movement is the element's current distance from the center point multiplied by the percentage value set on the slider.	

Step	Action	See also
	In the Converge layout, the element moves towards the center point; in the Diverge layout the element moves further away from the center point. Set the slider to the required percentage.	

5.5.7.8 Auto Route Layout

The *Auto Route* layout orthogonally routes connectors between the selected elements. The layout attempts to find the shortest path between the two connected elements while minimizing crossings. In the following layout, the original connectors are shown in red.



How to

To invoke the Autoroute layout

Step	Action	See also
1	Follow the general <i>Layout Diagrams</i> procedure, and at step 3 select Auto Route .	Layout Diagrams 769
2	When calculating connector routes, the algorithm divides the diagram into cells of a size determined by the Cell Size value; a smaller cell size results in connectors being placed closer together.	


Step	Action	See also
	In the Cell Size (px) field, type the value in pixels.	
3	In the Element Margin field, type the preferred separation between connector segments and element borders, in pixels	

5.5.7.9 Lay Out a Diagram Automatically

Enterprise Architect provides the facility to layout diagrams automatically. This creates a tree-based structure from the diagram elements and relationships in a diagram. Owing to the complexity of many diagrams, you might then have to do some manual 'tweaking'.

How to

To layout a diagram

Step	Action	See also
1	Select a diagram.	
2	Click on either: <ul style="list-style-type: none"> The Diagram Layout Diagram option, or The Auto Layout button on the diagram toolbar 	Layout Diagrams 

Notes

- This facility is available for Structural diagrams and Extended diagrams, but not for Behavioral diagrams (see the *UML Diagrams* topic for a description of the diagram types); however, the facility is also available for Sequence diagrams generated by the Enterprise Architect Debugger
- Dynamic and Analysis diagrams are **NOT** suited to this form of layout - please ensure first that the diagram type you are laying out benefits from the action
- If you dislike the autolayout, you can reverse it before saving the diagram; click (**Ctrl+Z**)

For a fine degree of control of the elements in your diagram, you can use the Diagram Layout Options dialog. Generally the default layout parameters provide adequate layouts for a wide range of diagrams, but there are times when more specific settings are required.

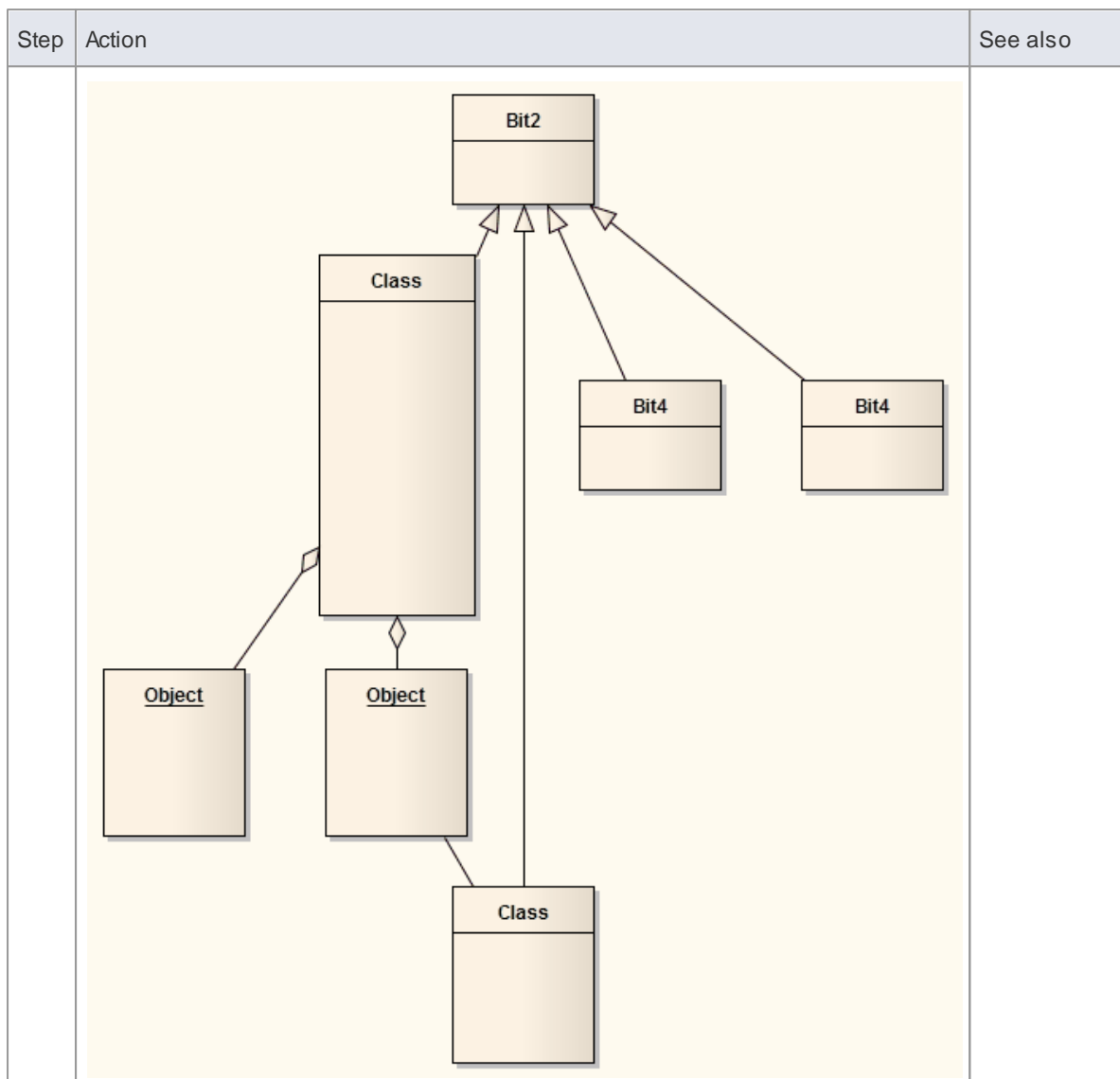
How to

To access the Diagram Layout Options dialog

Step	Action	See also
1	<p>Double-click on the background of the diagram.</p> <p>The Diagram Properties dialog displays.</p>	
2	<p>Click on the Diagram tab, then click on the Set Layout Style button.</p> <p>The Diagram Layout Options dialog displays.</p>	
3	<p>When you have made the required changes, click on the OK button to save the changes.</p> <p>You can alter any of the following settings on the Diagram Layout Options dialog to refine your layout:</p> <ul style="list-style-type: none"> • Cycle Remove Options panel - these settings remove cycles in the element organization (where element X is the source of a path but also becomes the target of a branch of the path), by reversing the connectors that impose the cycling and then reorganizing the diagram and reinstating the reversed relationships; this identifies the primary source element in the diagram <ul style="list-style-type: none"> • Greedy - Select to use the Greedy Cycle Removal algorithm, which minimizes the number of connectors reversed • Depth First Search - Select to use the Depth First Search Cycle Removal algorithm, which establishes the longest linear sequence possible, before establishing parallel sequences and branches; this algorithm is less effective in large and/or complex diagrams, but produces a more natural layout than the Greedy algorithm • Crossing Reduction Options panel - these options determine how long the routine should look for ways of reorganizing the layout to avoid crossed relationships: <ul style="list-style-type: none"> • Iterations - Type the number of iterations to be used during cycle removal (more than 8 does not usually provide any improvement) • Aggressive - Select to use an aggressive (detailed and time-consuming) crossing reduction step • Layering Options panel - these settings determine how elements are organized in layers during layout: <ul style="list-style-type: none"> • Longest Path Sink - Select to use the Longest Path Sink Layering algorithm, where the final target elements (sinks, which have no relationships issuing from them) are arranged in a layer at the top of the diagram, and the relationship paths built downwards from there in as many layers as there are nodes in the longest path 	

Step	Action	See also
	<ul style="list-style-type: none"> • Longest Path Source - Select to use the Longest Path Source Layering algorithm, where the original source elements (those with no relationships entering them) are arranged in a layer at the bottom of the diagram and the relationship paths built up from there in as many layers as there are nodes in the longest path • Optimal Link Length - Select to use the Optimal Link Length Layering algorithm, which organizes the elements into whichever layers minimize the total source-to-sink relationship chain; in this layout you can have both source elements and sink elements at various levels of the diagram <p>Layout Options panel:</p> <ul style="list-style-type: none"> • Layer Spacing - Type the default number of logical units between layers of elements (vertical spacing) • Column Spacing - Type the default number of logical units between elements within a layer (horizontal spacing) <p>Direction panel:</p> <ul style="list-style-type: none"> • Up, Down, Left, Right - Select the direction in which directed connectors should point, to set the position of the primary source element and the overall flow of the diagram <p>Initialize Options panel - the autolayout routine inserts line waypoints and connectors into relationship paths to help plot the direction of relationships; the routine then assigns an index number to every node, such that nodes in the same layer are numbered left to right</p> <p>The settings in this panel determine how those index numbers are assigned</p> <ul style="list-style-type: none"> • Naive - Select to use the Naive Initialize Indices algorithm, which assigns index numbers to nodes as they are encountered in a sweep and tends to place all waypoints to the right of real nodes (and therefore long relationships between a small number of elements to the right of chains of short relationships between several elements) • Depth First Search Outward - Select to use the Depth First Out Initialize Indices algorithm, which assigns index numbers to nodes as they are encountered in a depth first search from source nodes outwards (and would therefore place longer relationship chains to the left of shorter chains, with the primary source node at the start of the diagram flow) • Depth First Search Inward - Select to use the Depth First In Initialize Indices algorithm, which also assigns index numbers to nodes as they 	

Step	Action	See also
	<p>are encountered in a depth first search, but from sink nodes inwards (and would therefore place longer relationship chains to the left of shorter chains, with the ultimate target node at the end of the diagram flow)</p> <ul style="list-style-type: none">• Set as Project Default checkbox - select this checkbox to apply the diagram layout settings to all diagrams in the project; if you later check this box and click on the OK button for a different diagram, the new settings override the settings saved earlier <p>The illustration below is an example of an automatically laid out diagram, with the following options set:</p> <ul style="list-style-type: none">• Depth First Search• Optimal Link Length• Depth First Search Outward• Direction - Up	



Learn more

- [UML Diagrams](#)^[1010]

5.5.8 The Quick Linker

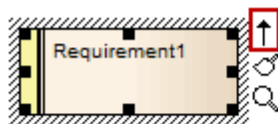
The *Quick Linker* provides a simple and fast way to create new elements and connectors on a diagram. When an element is selected in a diagram, the Quick Linker arrow is displayed at the upper right corner of the element, as shown below:

Topics

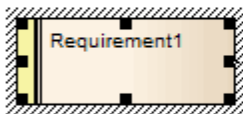
Image	Topic	Detail	See also
	Usage	Simply clicking and dragging the icon enables you to create new connectors and elements on a diagram, as explained in the following topics: <ul style="list-style-type: none"> • Create New Elements 	Create New Elements ^[792] Create Connectors ^[793]

Image	Topic	Detail	See also
		<ul style="list-style-type: none"> • <i>Create Connectors</i> <p>The connectors and elements suggested by the Quick Linker are the commonest objects appropriate to the context; you can select others from the Toolbox pages.</p> <p>Also, a Technology Developer can edit the lists of elements and connectors, and create new combinations.</p>	

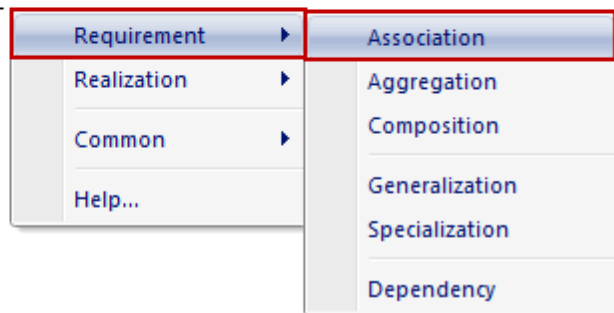
5.5.8.1 Create New Elements



Click, drag and release



Choose element and connector type

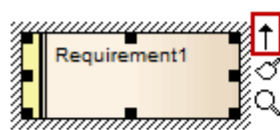


Topics

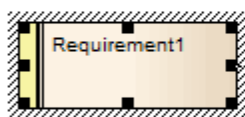
Topic	Detail	See also
Usage	<ul style="list-style-type: none"> • Press and hold (Shift) while selecting the type of connector to select an existing classifier as the target • For rapid modeling, you can suppress the Properties dialog when creating new elements; see the option Tools Options Objects Edit Object on New <p>The available Quick Linker options depend on the type of element selected; for example, the Quick Linker options for a Class differ</p>	

Topic	Detail	See also
	<p>from those of an Actor.</p> <p>These are the most appropriate, commonly used elements and connectors for the source element; you can create other target elements and connectors by selecting them from the appropriate Toolbox page.</p>	

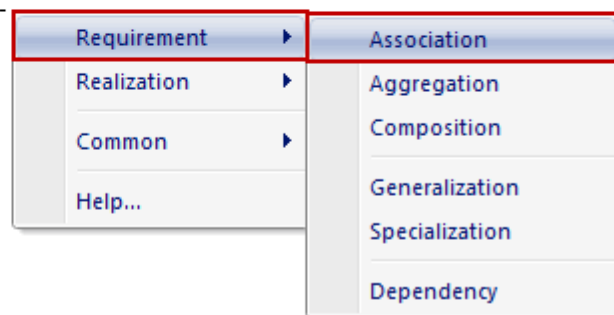
5.5.8.2 Create Connectors



Click, drag and release



Choose element and connector type



Topics

Topic	Detail	See also
Usage	<p>The list of connectors provides the most appropriate, commonly-used connectors for the source and target element types.</p> <p>If you want to use a different connector, select the appropriate Toolbox page, click on the required connector and then on the source element, and drag across to the target element.</p> <p>The connector does not actually establish until you release the mouse button over the target element; however, a dotted line shows where the connector would be at any point, and the solid outline of the nearest element or extension changes to a hatched outline as you move the cursor onto it - this helps you identify where the connector should connect to, if there are many closely-arranged elements, Parts, Ports and other extensions.</p> <p>You can also bend the connector, pressing (Shift) as you drag the cursor in a new direction.</p>	

Topic	Detail	See also
	You can create a Notelink to attach a Note or Constraint to a connector, simply by dragging the Quick Linker arrow on the Note to the connector.	

5.6 Elements



Models are constructed from elements, each of which has its own meaning, rules and notation. Elements can be used at different stages of the design process for different purposes.

The properties and features of an element can be defined and displayed through a set of windows and dialogs.

Learn more

- [Element Property Displays](#) ^[843]
- [Element Tasks](#) ^[795]
- [Element Context Menu](#) ^[827]
- [Visual Representation](#) ^[841]
- [In-place Editing Options](#) ^[916]

5.6.1 Element Tasks

This topic describes the following common tasks that you can perform on elements in Enterprise Architect:

Topic	Link
Create Elements	Create Elements ^[797]
Add Elements Directly to Packages	Add Elements Directly to Packages ^[798]
Use Auto Naming and Auto Counters	Use Auto Naming and Auto Counters ^[799]
Set Element Parent	Set Element Parent ^[801]
Show Element Use	Show Element Use ^[802]
Set Up Cross References	Set Up Cross References ^[803]
Move Elements Between Packages	Move Elements Between Packages ^[806]
Move Elements Within Diagrams	Move Elements Within Diagrams ^[805]

Topic	Link
Copy Elements Between Packages	Copy Elements Between Packages ^[808]
Change Element Type	Change Element Type ^[809]
Align Elements	Align Elements ^[809]
Resize Elements	Resize Elements ^[810]
Delete Elements	Delete Elements ^[811]
Customize Visibility of Elements	Customize Visibility of Elements ^[813]
Create Notes and Text	Create Notes and Text ^[814]
Link Note to Internal Documentation	Link Note to Internal Documentation ^[815]
Set an Element's Default Appearance	Set an Element's Default Appearance ^[817]
Use Element Templates	Use Element Templates ^[820]
Highlight Context Element	Highlight Context Element ^[822]
Make Linked Element a Local Copy	Make Linked Element a Local Copy ^[823]
Copy Features (Attributes and Operations) Between Elements	Copy Features (Attributes and Operations) Between Elements ^[823]
Move Features Between Elements	Move Features Between Elements ^[824]

Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Update Element permission to update

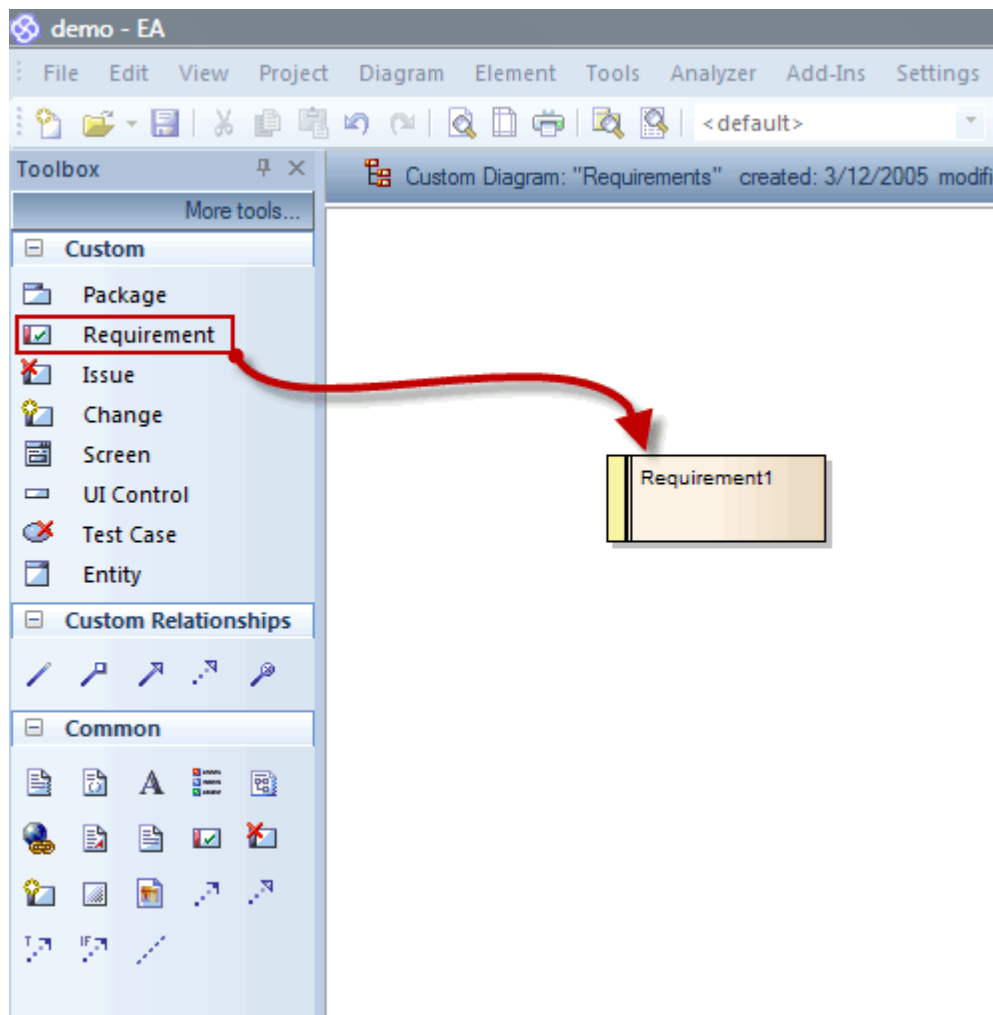
element properties or delete an element

Learn more

- [Permissions List](#)^[264]

5.6.1.1 Create Elements

Example



Notes

- If you have defined a template element in the Templates package, Enterprise Architect applies the characteristics of that template to each new element you create of that type
- If you are creating several elements of one type, after creating the first just press **(Shift+F3)** or **(Ctrl) +** click to create the next element of that type
- Be aware that once you have created elements, you can re-use them by dragging them from the Project Browser and dropping them onto your diagrams
- Sometimes it is useful to add elements to a package, without a diagrammatic representation; this can

be accomplished via the Project Browser and is explained in the topic - *Add Elements Directly to a Package*

Learn more

- [Create Elements In Place Using the Quick Linker](#)^[792]
- [Create Elements Using the Toolbox](#)^[698]
- [Create Elements Using the Diagram Context Menu](#)^[687]
- [Create Elements From Text File](#)^[1473]
- [Create a Group of Elements Using Design Patterns](#)^[1271]
- [Create Domain Specific Elements from UML Profiles](#)^[1277]
- [Pasting from the Tree](#)^[733]
- [Add Elements Directly to a Package](#)^[798]
- [Set Element Templates Package](#)^[820]
- [Behavioral Diagram Elements](#)^[1090]
- [Structural Diagram Elements](#)^[1179]

5.6.1.2 Add Elements Directly To Packages

You can quickly add new elements to a package without the necessity of adding a diagram element at the same time. This is particularly useful in defining a group of Requirements, Changes, Issues, base Classes or other element types that might not require diagrammatic representation in the model.

Access Project Browser package context menu | Add | Add Element

Reference

Complete the New Element dialog as described below:

Field/Button	Action	See also
Active Toolset	Shows the UML Diagram Type, profile, Add-In or MDG Technology within which you are creating the element.	
Select Toolset	<p>If the Active Toolset is not appropriate for the type of element you want to create, click on this button and select the required diagram type or technology.</p> <p>The <default> group in the list provides a basic set of elements drawn from across the UML Behavioral and Structural groups, and the Enterprise Architect Extended groups.</p> <p>The selected technology is displayed in the Active Toolset field.</p>	
Name	Type the name of the element.	
Auto	As an alternative to typing in the name, click on this button to insert	Use Auto

Field/Button	Action	See also
	<p>predefined auto-counter text.</p> <p>If you already have some text in the Name field, it is over-written by the auto-counter text.</p>	Naming and Auto Counters 799
Type	<p>Click on the drop-down arrow and select the element type from the list.</p> <p>The list of elements is derived from the diagram or technology shown in the Active Toolset field.</p>	
Stereotype	<p>If required, either type the stereotype name or click on the drop-down arrow and select the stereotype.</p>	
Open Properties Dialog on Creation	<p>Select the checkbox if the Properties dialog is to open immediately after the element is created.</p>	
Close Dialog on OK	<p><i>Deselect</i> the checkbox to add multiple elements in one session.</p>	
Add to current diagram	<p>Available if you have a diagram open.</p> <p>The checkbox is defaulted to selected to add the new element to the diagram; if you do not want the element in that diagram, deselect the checkbox.</p>	
Create	<p>Click on the OK button to create the element.</p>	

5.6.1.3 Use Auto Naming and Auto Counters

The Auto Element Naming dialog enables you to configure automatic naming for any element type. Each element type can have separately configured automatic names and aliases.

Access [Settings | Auto Names and Counters](#)

Reference

Use the Auto Name Counters dialog to set up auto naming:

Field/Option	Action	See also
Type	<p>Click on the drop-down arrow and select the element type to which to apply auto naming/auto numbering.</p>	
Name		

Field/Option	Action	See also
Prefix	Type a prefix for the new name (optional).	
Counter	Type the counter value; use as many 0's as required to pad the name.	
Suffix	Type a suffix for the new name (optional).	
Active	If required, click on the checkbox to turn auto naming on for the element type.	
Alias		
Prefix	Type a prefix for the new alias (optional).	
Counter	Type the counter value; use as many 0's as required to pad the alias.	
Suffix	Type a suffix for the new alias (optional).	
Active	If required, click on the checkbox to turn alias auto naming on for this element type.	
Save	<p>Click on the Save button.</p> <p>New elements of this type now have an automatically-generated name and/or alias with an incrementing counter value.</p> <ul style="list-style-type: none"> • If the user creates an element on a diagram, such as from the Diagram Toolbox or Quicklinker, the element is automatically assigned the defined name and/or alias • If the user creates an element outside a diagram (such as in the Project Browser) using the New Element dialog, they must click on the Auto button next to the Name and/or Alias field on the appropriate dialog to apply the auto-generated text • If the user already has some text in the Name or Alias field, it is over-written by the auto-counter text <p>If an Alias is active, to view the Alias in a diagram requires that the</p>	<p>Add Elements Directly To Packages ^[798]</p> <p>Diagram Properties ^[727]</p>

Field/Option	Action	See also
	option Use Alias if Available is selected in Diagram Properties.	

5.6.1.4 Set Element Parent

You can manually set an element's parent or parents, or an interface it realizes (implements), using the Set Parents and Interfaces dialog.

How to

To set an element parent

Step	Action	See also
1	Select a generalizable element in a diagram.	
2	Select the Element Advanced Set Parents and Interfaces menu option. Alternatively: <ul style="list-style-type: none"> • Press (Ctrl+I) or • Right-click and select the Advanced Parent context menu option. <p>The Set Parents and Interfaces dialog displays.</p>	
3	You can elect to enter a parent or interface name by either manually typing it in, or clicking on the Choose button to locate the element within the current model.	
4	Set the Type of relationship (Implements or Generalizes) from the drop-down list.	
5	Click on the Add button to add the relationship to the Type Details list, and to the selected child element. <div data-bbox="322 1615 584 1809" data-label="Diagram"> <p>The diagram shows a class hierarchy. A box labeled 'Book' is at the top. Below it is a box labeled 'Paperback'. A solid line with an open arrowhead points from 'Paperback' to 'Book', indicating that 'Paperback' is a specialization of 'Book'.</p> </div>	
6	Repeat this process to add further parent elements, or click on the Delete Selected button to remove a selected relationship.	

If the parent name is shown in the element on the diagram, you can select it and perform further operations, as follows:

Step	Action	See also
1	Click on the element.	
2	Click on the parent name to highlight it.	
3	<p>Right-click to display the context menu, which provides options enabling you to:</p> <ul style="list-style-type: none"> • Redisplay the Set Parents and Interfaces dialog to, for example, delete the parent or add further parents • Display the parent element Properties dialog • Find the parent element in the Project Browser <p>Find the parent element in the diagram.</p> <p>If the parent element does not currently exist in the model, and is listed on the element in the diagram, you can highlight it and click on it to redisplay the Set Parents and Interfaces dialog to, for example, delete it or add further parents.</p>	

Notes

- You can specify parent elements that exist in the model, or that you intend to create later; if the parent element does not currently exist in the model, select the **Accept classifier even if not in model** checkbox to ensure that the reference to a non-existent element is accepted
- If Parents are not in the same diagram as their corresponding related element, the parentage is shown in the top right corner of the child element

5.6.1.5 Show Element Use

You can display the use of an element using the Element Usage dialog. This lists all occurrences of the element throughout the model, and enables you to easily navigate to any occurrence.

How to

To show element usage

Step	Action	See also
1	Select an element in a diagram.	
2	Select the Element Find in Diagrams menu option (or, for a Package element, the Find	

Step	Action	See also
	<p>in all Diagrams option).</p> <p>Alternatively, press (Ctrl+U).</p> <p>If the element exists in other diagrams, the Element Usage dialog displays, listing all occurrences of the current element in diagrams in the model.</p>	
3	<p>If you want to display the usage information in a more readable layout, you can resize the dialog and its columns</p> <p>Either:</p> <ul style="list-style-type: none"> • Double-click on a line item to open the relevant diagram and display the selected element (the Element Usage dialog remains open), or • Click on the Open button to display the selected diagram and close the Element Usage dialog 	

Notes

- You can also access this feature from the Project Browser; select an element in the tree and select the **Element | Find in Diagrams** menu option

If there is only one instance of the element in any diagram, that diagram displays instead

5.6.1.6 Set Up Cross References

It is possible to set up a cross reference (or Custom Reference) from one element in Enterprise Architect to another. You can also view existing cross references on an element, using the Context References tab on the element's Properties dialog, or the Traceability window.

How to

To set up a cross reference

Step	Action	See also
1	In the Project Browser, locate the target element or diagram (that is, the object of the cross reference).	
2	Open a diagram that contains the elements that are to have the currently selected element as a reference.	
3	Right-click on the element in the Project Browser. The context menu displays.	
4	Select the Add custom reference menu option.	

Step	Action	See also
5	In the Set up references dialog, select the checkbox against each element to that is to have the target element as a reference.	
6	Optionally, in the Comment field, type some text to describe the purpose of the reference.	
7	Click on the OK button.	

To use the cross reference

Step	Action	See also
1	Select an element in a diagram.	
2	Select the Element Advanced Custom References menu option. Alternatively, either press (Ctrl+J), or right-click on the element and select the Find Custom References context menu option.	
3	The Custom References dialog displays, showing a list of elements that have been set as cross references for the selected element.	
4	You can open the Properties dialog for an element by highlighting it and clicking on the Open button.	
5	If you have a diagram cross reference, you can open that diagram.	
6	If you have a string of diagram links, click on the Home button to return to the original diagram.	

Notes

- You can delete a cross reference by selecting it on the Custom References dialog and clicking on the **Delete** button; cross references are also automatically deleted if the source or target element in the reference is deleted

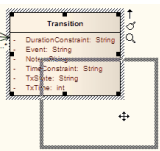
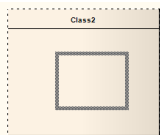
Learn more

- [Context References Tab](#)^[874]
- [The Traceability Window](#)^[637]

5.6.1.7 Move Elements Within Diagrams

Any one of the following options enables you to move an element within a diagram.

Topics

Image	Topic	Detail	See also
	Usage	<p>Select an element or group of elements in the diagram view, then:</p> <ul style="list-style-type: none"> • Use the mouse to drag the element to the required position (the cursor switches to the four-arrow icon as shown) • Hold down (Shift) and use the arrow keys to move the element by increments to the required position • Use the Left, Right, Up and Down options in the Element Move submenu • Align multiple elements using the Element Alignment submenu, the Alignment options in the right-click context menu, or the Alignment buttons on the Diagram toolbar 	
	Confirm Possible Parent Elements	<p>As you organize the elements within a diagram, you can drag any element over another and, provided the dragged element is within and on top of the possible parent, it is always encapsulated by the lower element and moved within the lower element.</p> <p>However, the lower element might not be a valid parent.</p> <p>You can confirm that a possible parent element is able to accept a selected child element. When you drag the child element over the potential parent, the target element border changes to a dashed line if it can accept the selected element as a child. If the border does not change, the selected element cannot be a child to the target element.</p>	

Notes

- The **Support for Composite Objects** checkbox must be selected on the Objects page of the Options dialog (select the **Tools | Options | Objects** option); if this option is not selected, the dashed border does not show and the child element cannot be embedded on the parent in the diagram

- Both elements must already exist on the diagram; an element border does not change if you drag a potential child element over it from the Toolbox or Project Browser
- The child element must have equal or higher Z-order placement than the parent; that is, the parent element must be level with or behind the child
- The child element borders must be completely within the parent element borders
- For example, if you drag a Signal over a Class, the Class border changes; a Class element can be a parent to a Signal. If you drag a Class element over a Signal element, the Signal border does not change. A Signal cannot be a parent to a Class
- When you embed a child element on its parent, the child element becomes part of the parent element hierarchy in the Project Browser. Similarly, if you drag the child element out of the parent, the child element becomes independent and is no longer embedded in the parent element hierarchy

Learn more

- [Objects](#) ^[556]
- [Diagram Context Menu](#) ^[687]

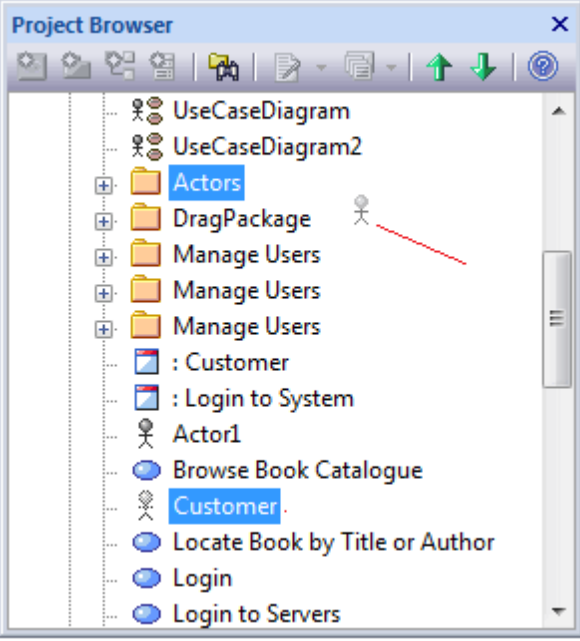
5.6.1.8 Move Elements Between Packages

Elements and packages can be moved from one package to another by dragging and dropping the element to the target destination in the Project Browser. Note that if you move a package, ALL the child packages and their contents are moved to the new location also.

How to

To move an element between packages

Step	Action	See also
1	Click on the element in the Project Browser. See <i>Customer</i> in the diagram below.	
2	Drag the element so that the cursor is over the target package icon. The element symbol (and, in some operating systems, the element name) displays at the moving cursor position.	

Step	Action	See also
		
3	<p>Release the mouse button.</p> <p>The element is moved into the target package.</p>	

Notes

- You can also drag the element under a host element in the new package; for example, drag an element under a Class
- Moving an element has no effect on any relationships that the element might have
- Moving an element in the Project Browser has no effect on the use of that element in any diagram
- Moving a diagram generally does not affect the location of elements in packages - if you move a diagram out of one package into another, all the elements in the diagram remain in the original package; however, certain elements (such as Decision, Initial and Final elements) are used only within one diagram, have no meaning outside that diagram, and are never re-used in any other diagram, therefore, if you move a diagram containing these elements, they **are** moved to the new parent package with the diagram
- In a multi-user environment, if one person moves or updates the Project Browser structure, other users must reload their project to see the latest changes in the Project Browser; although this is true of any addition or modification to the tree, it is most important when big changes are made, such as dragging a package to a different location

Learn more

- [Refresh View of Shared Project](#)^[242]

5.6.1.9 Copy Elements Between Packages

Enterprise Architect enables you to quickly and easily duplicate one or more elements, including their child elements and diagrams. You can insert a copy of an element under one or more other packages, in the same .eap file or any other .eap file.

Topics

Topic	Detail	See also
Copying Elements	<p>In the Project Browser, select each required element, right-click on one of them and select the Copy Element(s) to Clipboard context menu option.</p> <p>Alternatively, click on a selected element and press (Ctrl+C).</p> <p>The Copy Element(s) to Clipboard dialog briefly displays until the copy operation completes.</p>	
Pasting Elements	<p>In the Project Browser, right-click on the package into which to paste the copied elements, and select the Paste Element(s) from Clipboard context menu option.</p> <p>Alternatively, click on the package and press (Ctrl+V).</p> <p>The Paste Element(s) from Clipboard dialog briefly displays until the paste operation completes.</p> <p>The target package is expanded and the pasted elements are exposed in the Project Browser. If you are pasting the elements within the same model as the copied source, the source parent package is also collapsed.</p> <p>If the target package already contains:</p> <ul style="list-style-type: none"> • An element of the same type with the same name as a pasted element, the pasted element name has the suffix - <i>Copy</i> • An element with the same name as the pasted element including the - <i>Copy</i> suffix, the suffix becomes - <i>Copy1</i> (or - <i>Copy 2</i>, - <i>Copy3</i> and so on, as copies of the element accumulate in the target package) <p>You can keep the same element names as the source, or you can rename each element either by clicking twice on it and editing the name in the Project Browser, or by double-clicking on it and editing the name in the Properties dialog.</p>	

Notes

- A copy of an element does not have the external cross references of the source element; that is:
 - if one element is copied it has no connectors

- if more than one element is copied, only the connectors between the copied elements are retained
- however, if those elements come from a Sequence or Communication diagram and the diagram itself is not copied, the message connectors between the copied elements are not retained
- You cannot paste an element into a package that is locked by another user or that is checked in; the **Paste...** option is grayed out in the context menu

Learn more

- [Locking Packages](#) ^[273]
- [Checking In/Checking Out Packages](#) ^[353]

5.6.1.10 Change Element Type

Access **Element | Advanced | Change Type**

Select the new element type from the list and click on the **OK** button.

The target element is transformed into the required type.

5.6.1.11 Align Elements

How to

To align multiple elements

Step	Action	See also
1	Select a group of elements by drawing a selection box around them all (or select them one by one by holding down (Ctrl) and clicking on each element).	
2	Right-click on the element in the group to align others to. The context menu displays.	
3	Select the alignment function you require. All selected elements are aligned to the one beneath the cursor.	

Notes

- You can also use the Diagram toolbar; the first four buttons are used to align elements, and are made available when more than one element is selected in a diagram
- You can also select the **Element | Alignment** menu option, to display one of the **Position** submenus

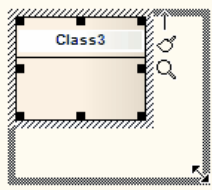
Learn more

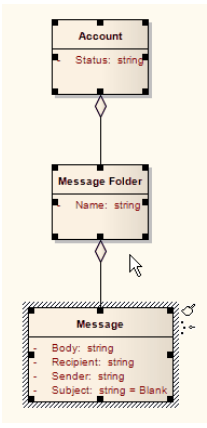
- [Diagram Toolbar](#)^[145]
- [Position Submenus](#)^[112]

5.6.1.12 *Resize Elements*

Any one of the following options enables you to resize an element.

Topics

Images	Topic	Detail	See also
	<p>Resize handles</p>	<p>Select an element or group of elements in the diagram view, then:</p> <ul style="list-style-type: none"> • Use the resize handles that appear at each corner and side to resize the element(s) by dragging with the mouse (the cursor switches to the double-ended arrow as shown in the image to the left) • Press and hold (Ctrl) and use the arrow keys to resize by increments as required • Use the Wider, Narrower, Taller and Shorter options in the Element Move submenu • Autosize selected element(s) using the option in the Element Appearance submenu, or by pressing (Alt+Z) (with multiple elements selected, Autosize also appears in the right-click context menu) • Set multiple elements to the same height, width or both, using these options in the Element Make Same submenu, or the options in the right-click context menu 	

Images	Topic	Detail	See also
	Resize a Set of Objects to a specific size	<p>If you right-click a selected set of objects, you can resize them to the same dimensions (height, width or both) using the context menu.</p> <p>When you select multiple elements using (Ctrl) +click, then resize the dimensions, the dimensions of the selected hatched object are used to set the dimensions of the other selected objects.</p> <p>For example, in the diagram to the left, the <i>Message Class</i> height and width are used to set the height and width of the <i>Account</i> and <i>Message Folder</i> Classes.</p> <p>The aim is to make the <i>Account</i> and <i>Message Folder</i> elements the same height and width as the <i>Message</i> element.</p>	

How to

To resize a set of objects to a specific size

Step	Action	See also
1	Set one element to the required size (for example, <i>Message</i> as above).	
2	Select all other elements (for example, <i>Account</i> and <i>Message Folder</i> as above).	
3	Right-click on the pre-sized element (for example, <i>Message</i>).	
4	Select your resizing option (such as same height or width) from the context menu.	

Learn more

- [Highlight Context Elements](#) ⁸²²

5.6.1.13 Delete Elements

How to

To delete an element from a *diagram*

Step	Action	See also
1	In the active diagram, click on the element to delete.	

Step	Action	See also
2	<p>Either:</p> <ul style="list-style-type: none"> • Press (Delete) or • Right-click to display the context menu and select the Delete <element name> option <p>This does not delete the element from the model, only from the current diagram.</p>	

To delete an element from the *model*

Step	Action	See also
1	<p>In the Project Browser, right-click on the element to delete and select the Delete <element name> context menu option.</p> <p>A confirmation prompt displays.</p>	
2	<p>Click on the Yes button.</p>	

Alternatively, click on the element in the Project Browser and press (**Ctrl+Delete**). The element is completely removed from the model.

To delete multiple elements from a *diagram*

Step	Action	See also
1	<p>In the active diagram, (Ctrl)+click on each element to delete.</p>	
2	<p>Either:</p> <ul style="list-style-type: none"> • Press (Delete), or • Right-click to display the context menu and select the Delete selected elements option 	

To delete multiple elements from a diagram *and* model

Step	Action	See also
1	Open the diagram containing the elements to remove from the model.	
2	Press (Ctrl+A) to select all of the elements in the diagram, or use (Ctrl)+click to select specific elements.	
3	Press (Ctrl+Delete) to completely remove the elements from the model.	

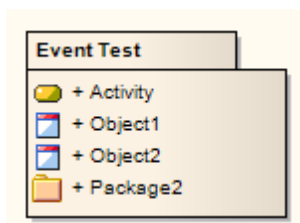
To delete multiple elements from the Project Browser and model

Step	Action	See also
1	In the Project Browser, press and hold either (Shift) or (Ctrl) and click on the required items.	
2	To completely remove the elements from the model, either: <ul style="list-style-type: none"> • Press (Ctrl+Delete), or • Right-click on the selected items and select the Delete selected item(s) context menu option <p>If you delete an element in this way, you delete all its properties and connectors as well.</p>	


5.6.1.14 Customize Visibility of Elements

Some elements are hidden from view in packages and in RTF documents by default. These include Events, Decisions, Sequence elements and Associations. You have the option of turning these elements back on.

For example, some Events and Decisions contained in a package do not appear in the package view, as in the example below.



How to

Images	Step	Action	See also
	1	Select the Tools Options Objects menu option. The Objects page of the Options dialog displays.	
	2	Click on the Advanced button. The Advanced Settings dialog displays.	
	3	Select the checkbox for each type of element to show in packages and in RTF documents.	
	4	Click on the Close button on each dialog.	
	5	Reload the current diagram if required. The package from the example shows the Event and Decision elements it now contains (see left image).	Refreshing the view of a shared project [242]

5.6.1.15 Create Notes and Text

You can create both notes and text on a diagram; the two are slightly different.

How to

To create a Note

Step	Action	See also
1	<p>Drag the Note icon from the Common page of the Toolbox onto the diagram.</p> <ul style="list-style-type: none"> If you have the Edit Object On New checkbox deselected on the Objects page of the Options dialog, the Note element displays on your diagram; type your note text directly within the Note element If you have the checkbox selected, the Notes <i>window</i> displays; type your text in that window <p>If you want to display the Notes information in a more readable layout, you can resize the window</p> <p>You can format the text if necessary, using the Notes toolbar at the top of the</p>	<p>Objects [556]</p> <p>Notes [977]</p> <p>Notes Toolbar [979]</p>

Step	Action	See also
	window; when you have completed the text, click on the OK button to save it	

To create a Text element

Step	Action	See also
1	<p>Drag the Text Element icon from the Common page of the Toolbox onto the diagram.</p> <p>The Notes window displays.</p>	Notes ^[977]
2	<p>Type your text in the window.</p> <p>If you want to display the Notes information in more readable layout, you can resize the window.</p> <p>You can format the text if necessary, using the Notes toolbar at the top of the window; when you have completed the text, click on the OK button to save it.</p>	Notes Toolbar ^[979]

Notes

- You can also create a note by clicking on the **New Note** icon (the text page) on the UML Elements toolbar and clicking on the diagram

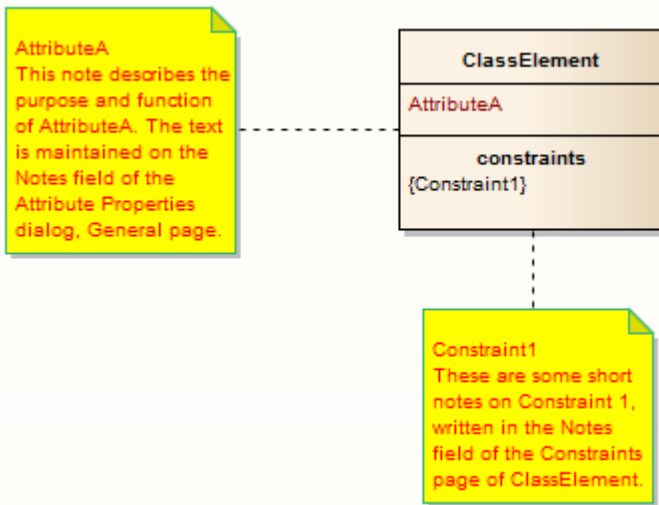
Learn more

- [UML Elements Toolbar](#) ^[143]

5.6.1.16 Link Note to Internal Documentation

It is possible to connect a *Note* element to another element's internal documentation. This enables you to externalize model documentation to the diagram level, and as Enterprise Architect keeps the Note and the internal structure in synch, you do not have to update the Note; in fact, you are blocked from editing the Note text, because it is maintained automatically.

In the example below, two notes are connected into an element's internal structures. One is connected to an attribute, and displays the attribute name and notes. The other is connected to a constraint, showing the constraint name and documentation.



How to

To connect a Note element to another element's documentation

Step	Action	See also
1	Click on the element and then click on the feature to link the Note to.	
2	Select the Element Inline Features Create Linked Note menu option. This creates a Note element linked to the selected feature, reflecting the content of that feature.	

Alternatively:

Step	Action	See also
1	Insert the target element into a diagram.	
2	Drag the Note icon from the Common page of the Toolbox onto the diagram, next to the target element. The Notes dialog displays.	
3	Do not type any text, just click on the OK button.	
4	Click on the Note Link icon from the Common page of the Toolbox.	

Step	Action	See also
5	Click on the Note, and drag across to the target element to create the connector.	
6	Right-click on the Note Link to display the context menu.	
7	Select the Link this Note to an Element Feature context menu option. The Link note to element feature dialog displays.	
8	In the Feature Type field, click on the drop-down arrow and select the type of feature to link to.	
9	In the Feature list, click on the specific feature to link to.	
10	Click on the OK button. The note now automatically derives its contents from the target element.	

Notes

- You can unlink a Note linked to an element feature, leaving the Note text, which can be edited manually and directly as normal; to unlink the Note from an element feature, either:
 - Right-click on the Note and select the **Advanced | Unlink from Element Feature** context menu option, or
 - Right-click on the Notelink connector, select the **Link this Note to an Element Feature** context menu option and, on the Link note to element feature dialog, set the **Feature Type** field to **None**

5.6.1.17 Set an Element's Default Appearance

To override the global appearance and define a default appearance for a specific element on all diagrams on which it is found, you use the Default Appearance dialog.

Access **Double-click element | Appearance | Default Appearance**

Use to

- Alter the background, border and font colors
- Change font styles, including font face and size
- Alter the border width for an element

Reference

Field	Usage	See also
Background Color	Set the color of the background for an element.	
Border Color	Set the color of the border for an element.	
Border Width	Set the width of the border surrounding an element, the minimum setting being 1 , the maximum setting being 5 .	
Font Style (...)	<p>Set the typeface of element text.</p> <p>The current font typeface is displayed within the text field; click on the (...) button.</p> <p>The Font dialog displays, which enables you to select a different typeface, style, (such as bold/italic) and font size.</p> <p>You can also select from a limited number of font colors and effects on this dialog.</p>	Font Dialog [839]
Font Color	Set the font color of the element text.	
Reset to defaults	Reset all appearance settings to default values.	
Preview	Review a representation of the colors, border width and font styles, which updates as you choose the new settings.	

Notes

- You can adjust several elements at the same time; select all of the required elements, right-click on one of them and select the **Default Appearance** context menu option, or use the Format toolbar
- If the **Show Project Custom Colors in Color Combo Boxes** checkbox is selected on the Options dialog, Standard Colors page, the last 16 color squares in the color selection palettes contain the project colors set on that dialog
- The **Font Color** field affects only the element name text; it does not override the feature text colors as defined on the Options dialog, Standard Colors page
- When it has been called directly on one or more selected elements (**Appearance | Set Font**), the Font dialog overrides the Default Appearance dialog
- When called from the Default Appearance dialog, the **Color** field on the Font dialog is overridden by the **Font Color** field on the Default Appearance dialog
- The **Font** and **Style** fields affect all text except for the Note compartment text
- The **Size** field affects all text

Learn more

- [Standard Colors](#)^[546]
- [Diagram Appearance](#)^[550]
- [Format Toolbar](#)^[692]

5.6.1.18 Get/Set Project Custom Colors

If more than one person is working on a project, you might want to share a set of custom colors with other team members, and possibly the whole project.

Access [Settings | Colors | Set Project Custom Colors](#)
[Settings | Colors | Get Project Custom Colors](#)

Use to

- Capture the project colors you have defined, in the project file, or
- Retrieve a set of colors someone else has defined, from the project file

How to

To set your project's custom colors in the project file

Step	Action	See also
1	Define the project colors on the Standard Colors page of the Options dialog.	Standard Colors ^[546]
2	Select the Set Project Custom Colors option. The following message displays: <i>Project Custom Colors export successful. 16 custom colors saved to the project file.</i>	
3	Click on the OK button.	

To get your project's custom colors from the project file

Step	Action	See also
1	Select the Get Project Custom Colors option. The following message displays: <i>Project Custom Colors import successful. 16 custom colors imported from the project file.</i>	
2	The colors are now made available in the color pallet of the Default Appearance	Set an

Step	Action	See also
	dialog.	Element's Default Appearance ^[817]

5.6.1.19 Set Element Templates Package

Topics

Topic	Detail	See also
Overview	<p>In building up a model, you might want to represent or emphasize certain characteristics of elements in the appearance of those elements, or select particular display options as standard.</p> <p>For example, you could make new Interface elements a different default color to new Class elements, ensure all new Activity Partitions are vertical rather than horizontal, or set a specific group of display options for new diagrams; you could also define a set of characteristics to use for each development stage of a project.</p> <p>To do all this, you create a diagram with all the characteristics you require, and store the diagram in an element Templates package; Enterprise Architect then checks this package:</p> <ul style="list-style-type: none"> • Whenever you start to create an element in a diagram or • Whenever you create a new diagram <p>If it finds a template for that diagram type, Enterprise Architect applies the settings in that template to the new element or to the display options of the diagram. For example, you could save a diagram under the name ClassTemplate, to apply a set of display characteristics to all new Class elements and all new Class or Logical diagrams.</p> <p>You should create the Templates package in an administrative View of the project file rather than in any work area, to prevent the package from being changed or lost in any project development work.</p>	
Modifying the appearance of elements	<p>There are two other ways in which you can modify the appearance of elements in diagrams:</p> <ul style="list-style-type: none"> • You can define the default appearance of elements (and other structures) grouped in a diagram by using UML Profiles; these provide a means of extending the UML Language, which enables you to build UML models in particular domains <p>Profiles are based on additional stereotypes and Tagged Values that are applied to elements, attributes, methods, connectors and connector ends</p>	UML Profiles ^[1277] UML Stereotypes ^[1269]

Topic	Detail	See also
	<ul style="list-style-type: none"> You can modify the appearance of elements (and connectors) of a specific type using stereotypes, which take precedence over templates; if you drop an unsteretyped element - a Class, for example - onto a diagram, Enterprise Architect searches the Templates package for a Class diagram that defines an unsteretyped Class, and applies that definition to the new Class <p>If you drop a stereotyped Class onto a diagram, the stereotype defines the Class appearance so the template is not accessed; stereotypes are much more flexible for defining the appearance of an element under different scenarios</p>	

How to

To set up the element Templates package

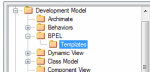
Image	Step	Action	See also
	1	<p>Create a new package in the appropriate administration View.</p> <p>You can give this package any name; <i>Templates</i> is an unambiguous option.</p>	
	2	<p>Within the Templates package create new diagrams, one for each type of diagram to template.</p> <p>Give them easily recognized names; for example <i>ClassTemplate</i> for the template for Class diagrams.</p>	
	3	<p>Add new elements to the template diagrams from the Toolbox, and configure the size, appearance, notes, version and other properties.</p>	
	4	<p>Select the Settings Project Template Package menu option to set the templates as the default element templates.</p> <p>The Browse Project window displays.</p>	
	5	<p>Locate and click on the Templates package, and click on the OK button to set the package as the default element template.</p> <p>Now each new element or diagram you add to your project is created with the settings from the appropriate Template diagram.</p>	

Image	Step	Action	See also

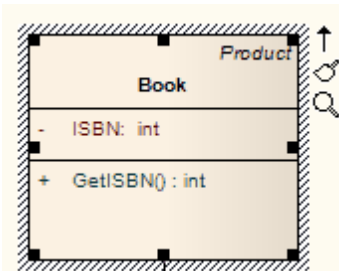
Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Reference Data - Update** permission to set up or access the element Templates package
- If you decide not to use the default element template, set the default element template to **<none>** in the Browse Project window; the **<none>** package is at the bottom of the hierarchy shown in the Browse Project window

5.6.1.20 Highlight Context Element

Access Tools | Options | Diagram | Behavior

You can show a hatched border around a selected element by selecting the **Always Highlight Context Element** checkbox on the Diagram Behavior page. If you have selected this checkbox, the selected element displays similarly to the following example:



If you have not selected the **Always Highlight Context Element** checkbox, the selected element does not have a hatched border around it.

Topics

Images	Topic	Detail	See also
	Multiple Selections	<p>Whether you have selected the Always Highlight Context Element checkbox or not, if you select multiple elements one of the elements you select always has a hatched border.</p> <p>If you align the elements, this element is the one used to align the other elements against.</p> <p>For example, if the elements in the diagram are aligned, the top element aligns to the bottom element (the element showing a hatched border).</p>	
	Change the	To change which element has a hatched border in	

Images	Topic	Detail	See also
	Element to Align Against	a selected group (and thus the element that is aligned with) click on the element that the other elements are to align with.	

5.6.1.21 Make Linked Element a Local Copy

How to

To convert a linked element to a local copy


Step	Action	See also
1	Open the diagram with the linked element.	
2	Select the linked element and right-click on it to display its context menu.	
3	Select the Convert Linked Element to Local Copy menu option. The element changes to a local copy and is placed in the appropriate package.	

5.6.1.22 Copy Features Between Elements

Using drag and drop, you can copy attributes and/or operations from an element in the Project Browser or Element Browser on to another element in a diagram.

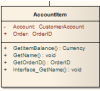
How to

To copy an element feature

Images	Step	Action	See also
	1	Open a diagram that contains the target element. In the example to the left, the <i>AccountItem</i> Class is the target and the <i>Customer</i> element is the donor - this is depicted for the Project Browser but is almost identical for the Element Browser.	
	2	Click on the attribute or operation and drag it to the target element.	
	3	Release the mouse button.	

The image below shows *AccountItem* after the attribute *Account* has been dropped from the browser on to it.

To copy multiple element features from the Project Browser (not the Element Browser)

Images	Step	Action	See also
	1	Open a diagram that contains the target element (as in the example above, the <i>AccountItem</i> Class is the target and <i>Customer</i> is the donor).	
	2	Hold down (Ctrl) (separate features) or (Shift) (select a range) and click on the attributes and/or operations to copy, then drag the selected features to the target element.	
	3	Release the mouse button.	

Learn more

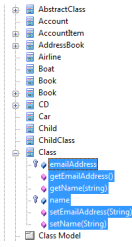
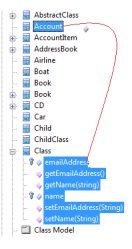
- [Attributes](#) ^[885]
- [Operations](#) ^[900]
- [The Project Browser](#) ^[567]
- [The Element Browser](#) ^[876]
- [Move Features Between Elements](#) ^[824]

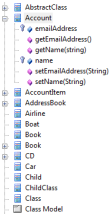
5.6.1.23 Move Features Between Elements

Using drag and drop, you can move attributes and/or operations from an element in the Project Browser on to another element within the Project Browser.

How to

To move element features

Images	Step	Action	See also
	1	In the Project Browser, locate the attributes and/or operations to move from the target element and select them while holding down: <ul style="list-style-type: none"> • (Ctrl) (single item select) or • (Shift) (multiple item select) 	
	2	Holding down the mouse button, drag the attributes and/or operations to the target element. A single feature symbol (and, under some operating systems, the feature name) displays during the move; however all of the selected features are moved.	

Images	Step	Action	See also
	3	<p>Release the mouse button.</p> <p>The image to the left shows the final stage of the attribute and operations move between the <i>Class</i> element and the <i>Account</i> element.</p>	

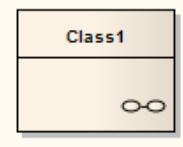
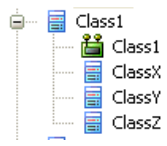
Learn more

- [Attributes](#) ^[885]
- [Operations](#) ^[900]
- [Copy Features Between Elements](#) ^[823]

5.6.1.24 Composite Elements

Enterprise Architect supports *Composite elements* for Classes, Objects and Use Cases. A Composite element is a pointer to a child diagram.

How to

Images	Step	Action	See also
	1	Create the element to set as a Composite element.	
	2	<p>Right-click on the element in the diagram and select the Advanced Make Composite context menu option.</p> <p>The element redisplay as illustrated on the left.</p> <p>Note the small icon in the bottom right hand corner indicating that this is now a Composite element.</p>	
	3	<p>Double-click on the Composite element to access the child diagram that it points to. You can populate the child diagram with elements and connectors to represent the internal structure of the (parent) composite element.</p> <p>At this point, embedded elements cannot be added to the child composite diagram. A modified procedure enables you to include them.</p> <p>The Composite element and its child diagram are represented in the Project Browser as shown.</p> <p>Note that <i>ClassX</i>, <i>ClassY</i> and <i>ClassZ</i> are elements added to the child diagram.</p>	Show Embedded Elements In Composite ^[826]

Topics

Topic	Detail	See also
Alternative Notation	<p>Composite elements can show their contents instead of their usual notation.</p> <p>To enable this:</p> <ol style="list-style-type: none"> 1. Right-click on the element to open the context menu. 2. Select the Advanced Show Composite Diagram option. <p>The element is replaced by a frame containing the child diagram. The frame label shows the element name.</p> <p>Some technologies, such as SysML, qualify the frame label to show the diagram type, element type, element name and child diagram name. If you design custom diagram profiles, you can also qualify the frame label as required.</p>	<p>Create Custom Diagram Profiles ^[1346]</p>
The Automation Interface	<p>Automation support is available for Composite elements.</p> <p><i>Element</i> has an <i>Elements</i> collection and a <i>Diagrams</i> collection.</p>	<p>Using the Automation Interface ^[2382]</p>

Notes

- If the **Make Composite** option is not listed in the context menu, the option is not available for the type of element you have selected

5.6.1.24.1 Show Embedded Elements In Composite

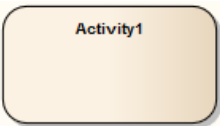
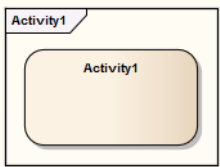
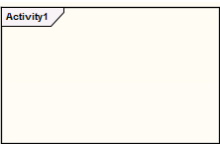
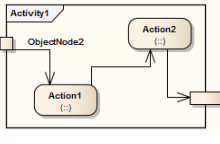
A Composite element shows the external relationships of the element with other components of the model and, via a child diagram, its own internal structure. You cannot add embedded elements to the initial representation of this *internal* structure; this procedure explains how to add embedded elements to the child diagram.

Embedded elements include:

- ActivityParameter
- RequiredInterface
- ActionPin
- Port
- EntryPoint
- ExpansionNode
- ObjectNode
- ProvidedInterface
- ExitPoint
- Part

How to

To include embedded elements in a child composite diagram

Images	Step	Action	See also
	1	Create the element to set as a Composite element. For example, a Class, Activity, or SysML Block.	
	2	Right-click on the element in the diagram and select the Advanced Make Composite context menu option. The element redisplay with a small icon in the bottom right hand corner, indicating that this is now a Composite element.	Composite Elements [825]
	3	Double-click on the Composite element to access the child diagram that it points to.	
	4	From the Project Browser, drag the Composite element itself onto the child Composite diagram, and paste it in as a Simple Link.	
	5	Right-click on the element in the diagram, and select the Advanced Show Composite Diagram context menu option. The element redisplay with a labeled frame around it.	
	6	Press (Ctrl+S) to save the diagram. Right-click on the diagram tab and select the Reload <diagram name> context menu option. The diagram redisplay with just the frame showing. This frame represents the original composite element.	
	7	Populate the frame with the appropriate internal elements, including the embedded elements that attach to the frame itself.	

5.6.2 Element Context Menu

Right-click on a single element in a diagram to open the element context menu. If two or more elements are selected, a different, multiple selection context menu is displayed.

The element context menu is split into a number of sections and sub-menus:

- Properties
- Add
- Find
- Transform (**Ctrl+H**) - Transform the selected element from one domain to another

- Embedded Elements
- Features
- Generate DDL - Generate DDL for a table, procedure or view Class
- Code Engineering
- Appearance
- UML Help - display the Enterprise Architect Help topic for the UML element type
- Delete (**Ctrl+D**) - delete the element

Notes

- Context menus vary between element types; for example, the Code Engineering options won't display for a Use Case element

Learn more

- [Properties](#) ^[828]
- [Add Sub-menu](#) ^[832]
- [Find Sub-menu](#) ^[833]
- [Transform Elements](#) ^[1683]
- [Embedded Elements](#) ^[834]
- [Features Menu](#) ^[835]
- [Generate DDL](#) ^[1978]
- [Code Engineering Menu](#) ^[836]
- [Appearance Menu](#) ^[837]
- [Multiple Selection Context Menu](#) ^[840]

5.6.2.1 Properties Menu Section

The **Properties** section of the element context menu can contain the following options:

Action	Usage	Shortcut	See also
Properties	Open the Properties dialog for the selected element. For State Lifeline and Value Lifeline elements, display the Configure Timeline dialog.	Ctrl + Enter	Properties dialog ^[843]
Advanced	Open the Advanced sub-menu.		Advanced Submenu ^[829]
Rule Composer	For a Rule Task element, invoke the Rule Composer tab in Business Rule Modeling.		Compose Business Rules ^[1536]





Action	Usage	Shortcut	See also
Other Properties	For State Lifeline and Value Lifeline elements, display the Properties dialog for the selected element.		Properties dialog [843]
Create (or Edit) Linked Document	(Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions) Create an RTF document linked to the element	Ctrl + Alt + D	Link Document to UML Element [929]
Delete Linked Document	Delete an existing linked document for the element.		

5.6.2.1.1 Advanced Submenu

The **Advanced** submenu on an element context menu can contain the options listed in the table below.

Action	Usage	Shortcut	See also
Parent	Set the element parent.	Ctrl + I	Set Element Parent [807]
Instance Classifier	Set the instance classifier for the element, on the Select <Item> dialog.	Ctrl + L	Select <Item> Dialog [880]
Classifier Properties	Open the Properties dialog for the <i>classifier</i> of the selected element.	Ctrl + Alt + Enter	Properties Dialog [843]
Make Composite	Set the element as a Composite element.		Composite Elements [825]
Change to State (Value) Lifeline	Switch one type of Lifeline element to the other.		
Show Composite Diagram	Display a mini-picture of the contents of a composite element within that element.		
Multiplicity	Define the multiplicity for the element, using the format defined on the Cardinality tab. This is the number of instances of the element that can exist in a set; the value displays on the element in a diagram, in the <i>Name</i> compartment.		Cardinality [986]

Action	Usage	Shortcut	See also
Show Element Type	(Ports and Parts only) Toggles between displaying and hiding the element type, as set on the Property page of the element's Properties dialog.		Part ^[1200] Port ^[1202] The Property Page ^[1204]
Edit Extension Points	For an extended Use Case, display the Use Case Extension Points dialog, which you use to insert the point at which the behavior should be inserted.		Use Case Extension Points ^[1176]
Association Class	Connect the Class to a new Association (if the element is a Class).		Connect New Class to Association ^[1216]
Use Rectangle (Circle) Notation	Use rectangle notation for the element.		Rectangle Notation ^[1177]
Partition Activity	Define an Activity Partition.		Activity Partition ^[1105]
Set Run State	Add a new instance variable to the element using the Define Run State dialog.		Define A Run-Time Variable ^[1198]
Set Property Value	(Part elements) Set the property value for the Part, using the Set Property Values dialog.		
Override Attribute Initializers	Pre-define initial values for attributes that can be used to override existing defaults.	Ctrl + Shift + R	
Convert to Instance (Property)	Convert this classifier to an instance or a property, depending on the type of classifier selected (for example, SysML classifiers are always converted to properties).		
Convert Linked Element To Local Copy	Convert the occurrence of the element on this diagram from a link to the original element to a local copy of the element.		
Make Sender/ Receiver	Toggle the element from a sender to a receiver and vice versa.		
Accept Time Event	Change the notation for an Accept Event action to an Accept Time Event action.		

Action	Usage	Shortcut	See also
Set Object State	Set the state of an object/instance based on the child states of its classifier.	Ctrl + Shift + S	Object State 
Define Concurrent Substates	Define a set of substates that can be held simultaneously within that composite State.		Regions 
Use State Label Notation	Display State Label Notation for a State object (the element name is displayed on a box on top of the element rather than inside it).		
Deep History	Change the type of a shallow History pseudo-state to a deep History. Applies only when right-clicking on a History pseudo-state.		History 
Set Attached Links	Attach the selected Note element to a connector, or several connectors.		Add A Note to A Connector 
Link to Diagram Note	Display the diagram notes as the Note element text. The option simply deletes any current text and blocks the Note from being edited other than through the Notes field in the diagram Properties dialog.		

Notes

- Context menus vary between element types. Not all menu options shown here are present on all element context menus; for example, the Partition Activity option only displays for an Activity element
- If an element appearance is overridden by a Shape Script, several of the appearance options are disabled; for example, **Use Rectangle (Circle) Notation**

Learn more

- [Shape Scripts](#) 

5.6.2.2 Add Submenu

The **Add** submenu enables you to add supporting elements and diagrams to the selected element.

Menu Option	Action	Shortcut	See also
Tagged Value	Add a Tagged Value.		Quick Add - Tagged Value To Elements [97]
Related Elements	Open the Insert Related Elements dialog.		Insert Related Elements [833]
Note	Create and attach a blank Note element to the current element.		Create Notes and Text [814]
Constraint	Create and attach a blank Constraint element to the current element.		Note [1147]
Activity	Add an Activity element as a child of the current Classifier element, with either an Activity diagram or an Interaction Overview diagram.		Activity [1101] Object Classifiers [898]
Interaction	Add an Interaction element as a child of the current Classifier element, with either a Sequence diagram, a Communication diagram or a Timing diagram.		Interaction [1137]
State Machine	Add a State Machine element as a child of the current Classifier element, with a State Machine diagram.		State Machine [1161]
RuleFlow activity	For a Class element, create a Rule Flow Activity element with a child Rule Flow diagram, as a behavior for the Class.		Create a Rule Flow Model [1530]
Add Diagram	Add a child diagram to the Classifier element, using the New Diagram dialog.		Add New Diagrams [724]

Notes

- Context menus vary between element types, so not all menu options shown here are present on all element context menus; for example, the options relating to Classifiers are not available for Object elements

5.6.2.2.1 Insert Related Elements

Access Right-click on element | Add | Related Elements

The Insert Related Elements dialog enables you to insert connected elements from elsewhere in the model into the current diagram.

You can specify the following details:

Field	Usage	See also
Insert elements to: «x» levels	Select the level down to which to insert connected elements, between levels 1 and 5 . You can select levels 4 or 5 to see how far the element/relationship hierarchy extends, but as this can produce a complicated and tangled diagram, it is better to use level 1 or 2 on selected elements in turn.	
For Link Type	Select a type of connector to limit the inserted elements to those connected by that relationship type.	
With Link Direction	Select whether the connectors are to be a single direction or bi-directional.	
Limit to Element Type	Select a type of element to limit the inserted elements to those of that element type.	
Layout Diagram When Complete	Select whether Enterprise Architect should layout the diagram after the elements have been inserted; the layout applied is the Digraph [778] layout. If no elements have been added, this option has no effect; elements have to be added for Enterprise Architect to adjust the layout.	Digraph Layout [778]
Limit to this Namespace	Select a specific namespace from which the inserted elements are to come.	

5.6.2.3 Find Submenu

The **Find** submenu on the element context menu can contain the following options:

Action	Usage	Shortcut	See also
In Project Browser	Highlight the currently selected element in the Project Browser.	Alt + G	

Action	Usage	Shortcut	See also
Locate Classifier In Project Browser	Highlight the classifier for the currently-selected object, in the Project Browser.	Ctrl + Alt + G	
Locate Operation in Project Browser	Highlight the call operation for the currently-selected Activity, in the Project Browser.	Ctrl + Alt + G	Class Operations in Activity Diagrams ^[1100]
In Diagrams	Open the Element Usage dialog.	Ctrl + U	Show Element Use ^[802]
Custom References	Set up cross references.	Ctrl + J	Set Up Cross References ^[803]
Add to Favorites	Add the element to the Favorites folder in the Resources window.		Favorites ^[1005]

5.6.2.4 Embedded Elements Submenu

The **Embedded Elements** submenu on the element context menu can contain the following options:

Menu Option	Action	Shortcut	See also
Add Port	Add an embedded Port to the element.		
Add Required Interface	Add an embedded Required Interface to the element.		
Add Provided Interface	Add an embedded Provided Interface to the element.		
Add Action Pin	Add an embedded Action Pin to the element.		
Add Expansion Node	Add an embedded Expansion Node to the element.		
Add Object Node	Add an embedded Object Node to the element.		
Add Activity Parameter	Add an embedded Activity Parameter to the element.		
Add Entry Point	Add an embedded Entry Point to the element.		
Add Exit Point	Add an embedded Exit Point to the element.		

Embedded Elements	Open the Embedded Elements window.		Embedded Elements Window [835]
Show Realized Interfaces	Display each interface directly realized by a Class.		Show Realized interfaces of Class [760]
Show Dependent Interfaces	Display each dependency relationship for that model element as a lollipop style node attached to its left-hand side.		

Notes

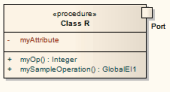
- Context menus vary slightly between element types, and not all menu options shown here are present on all element context menus; for example, of the **Add** options, only **Add Port** displays for a Class element

5.6.2.4.1 Embedded Elements Window

The Embedded Elements dialog enables you to embed particular elements into other elements. For example, a Port can be embedded into a Class.

Access [Diagram element context menu](#) | [Embedded Elements](#) | [Embedded Elements](#)

Topics

Image	Topic	Detail	See also
	Usage	<p>In the Embedded Elements dialog, click on the New button to create a new embedded element.</p> <p>Enter details such as type, name and stereotype, and click on the OK button.</p> <p>The embedded element now shows on the primary element as shown on the left.</p> <p>You can add as many embedded elements as necessary; modify or delete embedded elements using the Embedded Elements dialog.</p> <p>To incorporate inherited or owner properties, select the Show Owned/Inherited checkbox.</p> <p>The name of the embedded element is a label, which you can edit using the Labels context menu.</p>	Label Menu Section [760]

5.6.2.5 Features Menu Section

The **Features** section of the element context menu can contain the following options:

Action	Usage	Shortcut	See also
Attributes	Open the Attributes [885] dialog.		Attributes [885]
Operations	Open the Operations [900] dialog.		Operations Dialog - General [901]
Feature Visibility	Open the Feature Visibility [629] dialog.	Ctrl + Shift + Y	Feature Visibility [745]

Notes

- Context menus vary slightly between element types, and not all menu options shown here are present on all element context menus; for example, the **Attributes** and **Operations** options won't display for an Action element

5.6.2.6 Code Engineering Menu Section

The **Code Engineering** submenu on the element context menu can contain the following options:

Action	Usage	Shortcut	See also
Generate Code	Generate source code for the selected element (forward engineer).	F11	Generate a Single Class [1770]
Synchronize With Code	Reverse engineer source code for the selected element. You can also synchronize Class elements from the source code via the Project Browser, and reverse engineer or forward engineer model packages and code to synchronize them and incorporate changes made in one but not in the other.	F7	Importing Source Code [1790] Element Menu - Project Browser [577] Operation Menu - Project Browser [582] Update Package Contents [1773]
View Source Code	Open the source editor if a file exists for that selected element.	F12	Editing Source Code [1799]
Create Workbench Instance	Create a workbench instance for the Debug Workbench (if a debug command has been configured for the parent package).	Ctrl + Shift + J	Create Workbench Variables [2177]

Notes

- Context menus vary slightly between element types, and not all menu options shown here are present on all element context menus; for example, **Code Engineering** options won't appear for a Use Case element

5.6.2.7 Appearance Menu Section

The **Appearance** section of the element context menu can contain the following options:

Reference

Menu Option	Action	Shortcut	See also
Lock Element	<p>Lock the element so it can't be edited; to unlock the element, select Lock Element again.</p> <p>This does not apply in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions when security is enabled; in that situation, use the Lock option for model elements.</p>		Lock model elements ^[27]
Selectable	<p>Toggle whether the element is selectable or not</p> <ul style="list-style-type: none"> If an element is selectable, you can move it around the diagram and perform right-click context menu operations If an element is unselectable, you cannot move it around the diagram and the only right-click operation available is to make the element selectable <p>This option has no effect on double-click operations on the element, such as displaying child diagrams or Properties dialogs.</p> <p>An element on a locked diagram is also unselectable - if you click on it, the element outline displays in red.</p> <p>In a diagram, you can filter the display to show selectable or non-selectable elements only.</p>		Diagram Filters ^[629]
Dockable	<p>Align and join two elements either vertically or horizontally, on the current diagram only.</p> <p>Both elements must have the Dockable option selected, and must have the joining edges parallel.</p> <p>As the distance between the elements narrows,</p>		Partition ^[1148]

Menu Option	Action	Shortcut	See also
	<p>the moving element snaps to the edge of the other element.</p> <p>For Activity Partitions, the option is selected by default.</p> <p>Deselecting the Dockable option does not separate the elements; if necessary, you can simply move the elements apart again.</p>		
Appearance	Display the Appearance submenu; see the table below.		
Z-Order	Set the Z-Order of the element.		Z-Order Elements [744]

Appearance Sub-Menu:

Action	Usage	Shortcut	See also
Default Appearance	<p>Override the <i>global</i> default appearance of all elements (which you set on the Options dialog, Standard Colors page and Diagram Appearance page) with a different default for just the selected element on all diagrams in which it is found.</p> <p>To change the appearance of the selected element on the <i>current diagram only</i>, use the Format toolbar.</p>	F4	Standard Colors [546] Appearance [550] Set an Element's Default Appearance [817] Format toolbar [692]
Apply Image From Clipboard	<p>Paste the image held on the clipboard onto the selected element, as an alternative image.</p> <p>This image is added to the Image Library under an automatically-assigned name. The image can be accessed from and renamed in the Image Manager.</p>		Using the Image Manager [755]
Select Alternate Image	Select an alternative image using the Image Manager.	Ctrl + Shift + W	Using the Image Manager [755]
Hide/Show Name Under Image	Hide or redisplay the name label under an element with an alternative image.		
Set Font	Change the font type, size, color and effects for the text in an element.		Set Element Font [839]

Action	Usage	Shortcut	See also
Show Labels	Reveal any hidden labels on the element.		
Copy Appearance to Painter	Copy the default element appearance (set using the Default Appearance option, above) to the painter. You then paste the default appearance using the Paste Appearance option on the Diagram toolbar.		Diagram toolbar [145]
Copy Image to Clipboard	Copy the element image to the clipboard.		

Notes

- You can also change the appearance (and other aspects) of several selected elements at once
- Context menus vary slightly between element types, and not all menu options shown here are present on all element context menus; for example, the **Alternate Image** option won't display for a Lifeline element

Learn more

- [Element Multiple Selection Menu](#) [840]

5.6.2.7.1 Set Element Font

Topics

Topic	Detail	See also
Usage	<p>You can change the appearance of the text within an element, for one or more selected elements, by:</p> <ul style="list-style-type: none"> • Selecting the Appearance Set Font context menu option • Selecting the Font icon on the Format toolbar, or • Selecting the Font Style option on the element Default Appearance dialog (for the selected element, overridden by the two options above) <p>The Font dialog displays.</p> <p>Select the font, style, size, color and effects, and (if necessary) the script type.</p> <p>Click on the OK button to save your changes.</p>	Format Toolbar [692] Default Appearance [817]

5.6.2.8 Element Multiple Selection Menu

You can perform operations on two or more elements on a diagram at once. To select the required elements, either click and drag the cursor over the group to highlight them, or press (**Shift**) and click on each element.

Topics

Topic	Detail	See also
Usage	<p>Right-click on an element to display a context menu that enables you to:</p> <ul style="list-style-type: none"> Align elements (by left edge, right edge, top, bottom, center in a column or center in a row) Space elements evenly (across or down) Standardize the dimensions of the selected elements Specify the visibility of features for all selected elements Add the same Tagged Value to all selected elements Automatically resize elements to match (element content permitting) Turn the Dockable option on or off for all selected elements on a diagram Set the default appearance and font for multiple elements at once Make the selected elements on the diagram non-selectable; to make them selectable again, right-click on the diagram and select the Make All Elements Selectable context menu option Generate code for all selected elements at once, or synchronize the code against the selected elements Transform the selected elements Copy all selected elements to the clipboard Automatically adjust the layout of the selected elements on the diagram Delete all selected elements 	<p>Feature Visibility [745]</p> <p>Assign a Tagged Value to an Item [972]</p> <p>Appearance Menu Section [837]</p> <p>Set an Element's Default Appearance [817]</p> <p>Set Element Font [839]</p> <p>Transform Elements [1683]</p>

Notes

- Where elements are made the same, they are matched to the element you right-clicked on
- It is much faster to assign an appearance or characteristic to a group of elements than to one element at a time

5.6.3 Visual Representation

Each UML element has a default representation. However, you can:

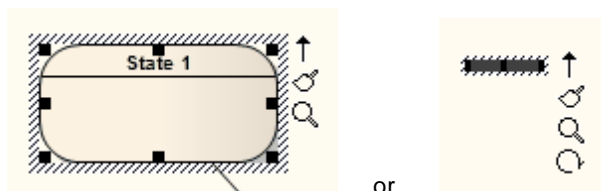
- Change the appearance, features, position and - for some elements - orientation through a number of toolbars
- Display or hide various types of information held in the element, in compartments

Learn more

- [Element Icons](#) ^[841]
- [Compartments](#) ^[842]

5.6.3.1 Element Icons

When you add an element to a diagram, or select an existing element, a number of small icons display off the right hand side of the element, underneath the Quicklinker arrow.



These icons display small versions of the diagram toolbars or perform specific actions, to enable you to quickly edit the element you have highlighted.

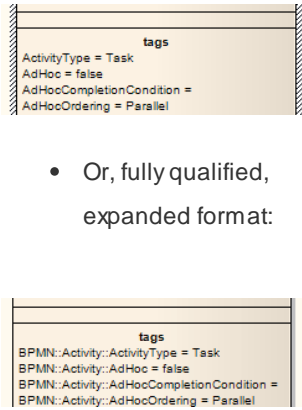

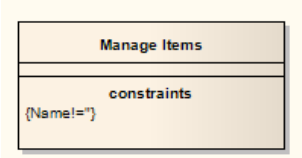
Reference

Icon	Description	See also
	Rotates a Fork/Join element from vertical to horizontal and vice-versa.	
	Displays the Format toolbar, for changing element appearance.	Format Toolbar ^[692]
	Displays the Current Element toolbar, to edit the element's properties and features.	Current Element Toolbar ^[146]
	When multiple elements are selected, displays the Diagram Toolbar for changing or aligning the elements together.	Diagram Toolbar ^[145]
	Toggles an Activity Partition between vertical and horizontal.	Activity Partition ^[1105]

5.6.3.2 Compartments

In addition to the attributes and operations compartments shown in a Class element, Enterprise Architect enables you to display or hide other compartments.

Topics

Images	Topic	Detail	See also
 <ul style="list-style-type: none"> Or, fully qualified, expanded format: 	Tags Compartment	<p>The <i>tags</i> compartment lists all Tagged Values for an element as entered in the Tagged Values window.</p> <p>The fully-qualified option operates only on those Tagged Values that were created in Enterprise Architect release 7.1 or later; it does not expand Tagged Values created in earlier releases.</p>	Tagged Values [969]
	Responsibilities Compartment	The <i>responsibilities</i> compartment shows a list of responsibilities as entered on the Requirements page of the element Properties dialog.	Requirements [848]
	Constraints Compartment	The <i>constraints</i> compartment shows a list of element constraints as entered in the Constraints page of the element Properties dialog.	Constraints [850]
	Testing Compartment	The <i>testing</i> compartment lists all of the tests associated with an element as listed in the Testing window.	The Testing Workspace [2213]
	Maintenance Compartment	The <i>maintenance</i> compartment lists all of the defects, changes, issues and tasks associated with an element, as listed in the Maintenance window.	The Maintenance Workspace [2237]

Notes

- To set the visibility of the various compartments, see the *Feature Visibility* topic

Learn more

- [Feature Visibility](#)^[745]

5.6.4 Element Property Displays

To define, assign and review the properties and features of elements, you can use the following:

Topic	Link
Properties Dialog	Properties Dialog ^[843]
Properties Docked Window	Properties Window ^[878]
Element Browser	The Element Browser ^[876]
Scenarios & Requirements Window	The Scenarios & Requirements Window ^[879]
Select <Item> Dialog	Select <Item> Dialog ^[880]

5.6.4.1 Properties Dialog

This topic area covers element properties and their settings, responsibilities, constraints, connectors, scenarios, Tagged Values, associated files, object files and classifiers, and boundary element settings.

Topics

Topic	Detail	See also
Displaying the Properties dialog	<p>To display the element Properties dialog, use any of the following methods:</p> <ul style="list-style-type: none"> • Select an element in the Diagram View and select the Element Properties menu option • Right-click on an element in the Diagram View, and select the Properties context menu option • Select an element in the Diagram View, and press (Alt+Enter) • Double-click on an element in the Diagram View • Right-click on an element in the Project Browser, and select the Properties context menu option 	Objects ^[556]

Topic	Detail	See also
	To suppress display of the Properties dialogue when placing a new element, uncheck the Edit Object on New option on the Objects page of the Options dialog.	
Properties dialog pages	<p>The Class Properties dialog consists of the following pages:</p> <ul style="list-style-type: none"> • General • Details • Advanced Properties • Templates • Tagged Values • Requirements • Constraints • Scenarios • Files • Links <p>Properties also include Object files and Classifiers, and the Boundary element appearance.</p>	<p>General Settings ^[845]</p> <p>Details ^[846]</p> <p>Advanced Properties ^[847]</p> <p>Templates ^[848]</p> <p>Tagged Values ^[969]</p> <p>Requirements ^[848]</p> <p>Constraints ^[850]</p> <p>Scenarios ^[852]</p> <p>Associated Files ^[875]</p> <p>Links ^[851]</p> <p>Object Classifiers ^[898]</p> <p>Boundary Element Settings ^[1171]</p>

Notes

- There are several variations of the Properties dialog:
 - The dialog for a Table or Stored Procedure element has slight differences on the General page, and a Table (Stored Procedure) Details page instead of a Details page
 - The dialog for a Class element of a stereotype other than Table is as shown in General Settings
 - The dialog for an element of any other type does not have a Details page
 - Port and Part elements have a Property page
 - Activity elements have a Behavior page, and Action and Invocation elements (depending on their type) have other pages such as Effect, Trigger and Call pages
 - Action Pins have a Pin page
- In all cases, the Properties dialog is an expandable window, which you can stretch to enable longer entry and clearer inspection of the text field values
- The Tagged Values page of the element Properties dialog simply provides the Tagged Values window within the frame of the Properties dialog.

Learn more

- [Working With Table Properties](#) ^[1937]

- [Port](#)^[1202]
- [Part](#)^[1200]
- [The Property Page](#)^[1204]
- [Interactions and Activities](#)^[910]
- [Action](#)^[1091]
- [Trigger](#)^[1173]
- [Behaviour Calls](#)^[910]
- [Action Pin](#)^[1097]

5.6.4.1.1 General Settings

The General page of the element Properties dialog contains the following fields:

Field	Usage	See also
Name	Change the element's name.	
Stereotype	(Optional) Type the name of a stereotype for the element, or click on the drop-down arrow and select one.	
Role	Type the role played by the object represented by the element.	
Alias	Type an alias (alternative display name) for the object.	
Keywords	Enter free-text items such as keywords or context information; this can be filtered in Use Case Metrics and Search dialogs.	
Author	Enter or select the name of the original author.	
Language	Select the programming language for the object.	
Status	Indicate the current status of the element (such as Approved or Proposed).	
Complexity	Indicate the complexity of the element (used for project estimation), such as Easy, Medium or Hard.	
Version	Enter the version of the current element.	
Phase	Indicate the phase this element is to be implemented in (for example, 1, 1.1, 2.0, ...).	

Field	Usage	See also
Notes	Enter any notes text associated with the element, as described for the Notes window, You can format the notes text using the Notes toolbar at the top of the field,	Notes ^[977] Notes Toolbar ^[979]

Further facilities are made available by clicking on the Advanced page.

Learn more

- [Advanced Settings](#) ^[847]

5.6.4.1.2 Details

The Details page of the element Properties dialog enables you to define the structural and processing details for the selected Class element.

Reference

Field/Button	Usage	See also
Scope	Click on the drop-down arrow and select the element's scope (public, private, protected, package).	
Persistence	Click on the drop-down arrow and select the appropriate value (blank, persistent or transient).	
Cardinality	The number of instances of the element that can exist - the value displays on the element in a diagram, in the <i>Name</i> compartment. Use the format defined in the Cardinality tab.	Cardinality ^[986]
Abstract	Select if the element is abstract.	
Is Root	Select if the element is a root element and cannot be descended from another.	
Is Leaf	Select if the element is final and cannot be a parent for other elements.	
Is Specification	Select if the element is a specification.	

Field/Button	Usage	See also
Is Active	Select if the element is active; for example, an Active Class.	Active Classes ^[1182]
Attributes	Define attributes for the Class. The Attributes Properties dialog displays.	Attributes ^[885]
Operations	Define operations for the Class. The Operations Properties dialog displays.	Operations Dialog - General ^[901]
Concurrency	Specify how concurrent activities should be processed.	
Collection Classes	Define Collection Classes (for generating code from Association connectors) that apply to this Class. The Collection Classes for Association Roles dialog displays.	Set Collection Classes ^[1904]
Template Parameters	List the <i>Parameterized Class</i> template parameters. Click on the Edit button to edit a selected parameter, or click on the Add button to add a new parameter.	Parameterized Classes ^[1183]
Binding(s)	List the binding expressions on a <i>binding Class</i> . Click on the Edit button to edit a selected binding expression, or click on the Add button to add a new binding expression.	Template Binding ^[1257]

Notes

- When launched from MDG Integration, the **Attributes** and **Operations** buttons are not available

5.6.4.1.3 Advanced Properties

The Advanced page displays the advanced properties of the element (where they exist) and enables you to reset the values of these properties.

How to

To set a value for an advanced (or custom) property

Step	Action	See also
1	Click on the data field to the right of the property name.	

Step	Action	See also
2	Depending on the property, either: <ul style="list-style-type: none"> • Type the value in free text • Click on the drop-down arrow and select the value from the list • Click on the (...) Browse button and search for the required value 	
3	Click on the Apply button.	

5.6.4.1.4 Templates

Use the Templates page to define

- Parameterized Class template parameters
- Binding expressions on a binding Class

In the Template Parameter(s) panel, click on the **Add** button to add a new parameter, or click on the **Edit** button to edit a selected existing parameter; in either case, the Template Parameter dialog displays.

In the Binding(s) panel, click on the **Add** button to add a new binding expression or click on the **Edit** button to edit a selected binding expression. When you click on the **Add** button, a short context menu displays from which you select the type of relationship to generate, and then you define the binding expressions.

If either a template parameter or a binding expression is no longer required, click on it and then click on the corresponding **Delete** button.

Learn more

- [Parameterized Classes \(Templates\)](#)^[183]
- [Template Binding](#)^[257]

5.6.4.1.5 Requirements

You use the Requirements page of the element Properties dialog to create requirements that this element is designed to meet.

Requirements are of two types:

- Internal requirements (responsibilities) and
- External requirements (system requirements, elements connected to the element by a Realize connector)

Enterprise Architect shows both types, but you can only edit the internal type from the Requirements page.

You can show the requirements for an element on the diagram directly, using the Feature Visibility function; it is also possible to show inherited requirements in this way.

Field/Button	Usage	See also
Requirement	Enter the name and high level detail of the requirement.	
Type	Specify the type; for example, Functional or Non-functional: <ul style="list-style-type: none"> • Functional requirements are things that the system must do, such as identify franked, unfranked and total credit for a dividend • Non-functional requirements are things that the system must be, such as reliable or cost effective 	
Last update	Specify the date of the last requirement update.	
Status	Specify the current status of the requirement.	
Difficulty	Identify the difficulty of implementing the current requirement.	
Priority	Specify how urgent the requirement is.	
Stability	Specify the estimated stability of the requirement. This is an indication of the probability of the requirement - or understanding of the requirement - changing; high stability indicates a low probability of the requirement changing.	
Notes	Record details of the requirement. You can format the notes text using the Notes toolbar at the top of the field.	Notes Toolbar ⁹⁷⁹
Move External	Make an internal responsibility into an external requirement.	Make Internal Requirement External ¹⁴⁸⁶
New	Create a new requirement.	
Save	Save changes to requirements.	
Delete	Delete a selected requirement.	

Field/Button	Usage	See also
Defined	List the defined requirements associated with this element.	

Learn more

- [Internal Requirements](#)^[1486]
- [External Requirements](#)^[850]
- [Feature Visibility](#)^[745]

5.6.4.1.5.1 External Requirements

External requirements are those Requirement elements that have been connected to the current element using a *Realization* connector. By creating the connector from the element to the requirement, you create an expectation that the element must implement the requirement as part of the system solution.

In Enterprise Architect, linked requirements are shown in the Requirements page of the element Properties dialog, but they are marked external and cannot be directly edited (on selection, the page fields are grayed out).

Double-click an external requirement in the list to activate the Properties dialog for the associated requirement, where you can view and modify the requirement details and check the requirement hierarchy details.

Learn more

- [Create Requirements](#)^[1473]
- [Requirement Properties](#)^[1475]
- [Make Internal Requirement External](#)^[1486]

5.6.4.1.6 Constraints

Elements can have associated constraints placed on them. Constraints are used in conjunction with responsibilities to define the conditions and rules under which an element operates and exists.

Typical constraints are pre- and post- conditions, which indicate things that must be true before the element is created or accessed and things that must be true after the element is destroyed or its action complete.

Use the Feature Visibility function to show constraints for an element on the diagram directly; it is also possible to show inherited constraints in this way.

How to

To add constraints to a model element

Step	Action	See also
1	Open the element Properties dialog.	
2	Select the Constraints page.	

Step	Action	See also
3	In the Constraint field, type the name of the constraint.	
4	In the Type and Status fields, click on the drop-down arrow and select the constraint type (Pre-condition , Post-condition or Invariant) and status.	
5	In the larger text field, type any additional notes required.	
6	Click on the Save button.	

Learn more

- [Feature Visibility](#)^[745]
- [Requirements](#)^[848]

5.6.4.1.7 Links

The Links page of the element Properties dialog displays a list of all relationships active for the current element.

Topics

Topic	Detail	See also
Relationships Panel	<p>The Relationships panel lists the relationships this element has, including:</p> <ul style="list-style-type: none"> • Elements this element is related to • Element Stereotype (if any) of the element • Type of the related element • Connection or relationship type • Stereotype (if any) of the relationship 	
Links Page	<p>From the Links page you can perform operations on a relationship, by right-clicking on the relationship to display the context menu.</p> <p>To:</p> <ul style="list-style-type: none"> • Hide the relationship on the diagram, click on the Hide Relation menu option; the option then changes to Show Relation, which you select to redisplay the relationship on the diagram • Display the relationship Properties dialog, click on the Relationship Properties menu option 	Connector Properties ^[961]

Topic	Detail	See also
	<ul style="list-style-type: none"> Highlight the related element in the Project Browser, click on the Locate Related Object menu option Delete the relationship from the model and all diagrams, click on the Delete Relationship menu option; the system prompts you to confirm the deletion 	

5.6.4.1.8 Scenarios

A scenario is a real-world sequence of operations that describes how an element works in real-time.



Topics

Topic	Detail	See also
Where you can generate Scenarios	<p>You can apply scenarios to any element - generally Use Cases - to describe functional behavior, business work flows and end-to-end business processes.</p> <p>You create scenarios through the Scenario page of the element Properties dialog; this page has two internal tabs, as described below.</p> <ul style="list-style-type: none"> The Description tab enables you to create scenarios and provide a simple text description of each scenario, or of the structure of each scenario The Structured Specification tab (the default) enables you to create scenarios or select those you have created elsewhere and, for each scenario: <ul style="list-style-type: none"> Create a series of steps for each part of the scenario Structure the scenario to show how the basic path diverges into the alternate paths and exception paths Generate a number of types of diagram from the structure Generate a structured scenario from an Activity diagram Generate a structured scenario from text on the clipboard; this option has a variation in the Description tab that enables you to translate scenario descriptions created prior to release 8.0 of Enterprise Architect, into structured scenarios in the 	<p>Structured Specification Tab ^[854]</p> <p>Generate Diagrams ^[864]</p> <p>Generate Scenario from Activity diagram ^[873]</p> <p>Structured Specification Item Context Menu ^[859]</p> <p>Scenarios ^[853]</p>

Topic	Detail	See also
	latest release	
Description Tab	<p>On the Scenarios page, click on the Description tab.</p> <p>The Scenario (name) field and the Type field both default to Basic Path, to enable you to define the basic path first; you can overwrite the scenario name with more appropriate text if required.</p> <p>As you go on to create other scenarios, you set the type to Alternate or Exception as appropriate.</p> <p>Complete the fields as described in the table below.</p>	

Reference

Field	Usage	See also
Scenario	Type in the name of the scenario (or, for existing scenarios, click on the drop-down arrow and select one from the list).	
Type	<p>Specify the type of scenario; the options are:</p> <ul style="list-style-type: none"> • Basic Path - the direct set of steps for the scenario • Alternate - an alternative set of steps, in parallel with part of the basic path • Exception - the path the scenario follows if a step of the basic path does not produce an appropriate result 	
Description	<p>Record a textual description of how the user uses the current element.</p> <p>As for the Notes window, you can format the notes text using the Notes toolbar at the top of the field.</p> <p>As well as the Notes window facilities, you can also generate a structured specification from the text in this field; highlight the text, right-click on it and select the Create Structure from Notes context menu option.</p> <p>The text is copied to the Structured Specification tab for the current scenario, either as a new specification or as the continuation of an existing specification, with a new step created at each carriage return.</p> <p>Subsequent changes to the text on the Structured Specification tab are not reflected on the Description tab.</p>	<p>Note Tab^[977]</p> <p>Notes Toolbar^[979]</p> <p>Structured Specification Tab^[854]</p>
New	Clear the data fields so that you can enter data in them to create a new scenario.	

Field	Usage	See also
Save	Save a new scenario, or changes to an existing scenario.	
Delete	Delete a scenario selected from the Scenarios panel, below.	
Scenarios	Display a list of defined scenarios for this element. You can change the order in which the scenarios are listed, using the  and  buttons.	

Notes

- The Scenario page does not prevent you from creating more than one basic path, but it would be unusual to define more than one
- All the functions available on the Scenario page are also available through the Scenarios & Requirements window/view; use the **Browse Element** icon in the window toolbar to list and select the scenarios for the element

Learn more

- [Scenarios & Requirements](#) ^[879]

5.6.4.1.8.1 Structured Specification Tab

The Structured Specification tab offers a wide range of facilities for generating and modifying scenario specifications, enabling you to define the structure, actions and interactions of the scenarios defined for an element such as a Use Case. These scenarios can be the main (basic) path, alternate paths or exception paths.

When you open the Structured Specification tab it defaults to the basic path so that if it does not already exist, you can create it. You can create alternate paths and exception paths as part of the process of adding them to steps of the basic path.

You can also create all three types of scenario paths on the Description tab, or in the Scenarios & Requirements window.

Learn more

- [Scenarios](#) ^[852]
- [Scenarios & Requirements](#) ^[879]
- [Structured Specification Toolbar](#) ^[857]
- [Item context menu](#) ^[859]
- [Selected Text context menu](#) ^[861]
- [Entry Points context menu](#) ^[863]

- [Floating Toolbar](#)^[864]

You can create a specification for a scenario in one of several ways:

- Enter the specification yourself, as described below
- Generate a specification from an Activity diagram created under a Use Case element
- Generate a specification from the notes text of the scenario in the Description tab
- Generate a specification from text held on the clipboard

How to

To enter the specification yourself, starting with the basic path

Step	Action	See also
1	In the Scenario field, click on the drop-down arrow and select the <i>Basic Path</i> scenario.	
2	In the <i>new step</i> field in the Action column, type the text of the first step or action.	
3	Tab to the Uses column and, if necessary, type the name of each element used in this step; any elements that are listed in the Context References tab are highlighted in blue and underlined. You can also manage the list of elements as context references, and add to the list directly; see the Notes below.	
4	Tab to the Results column and, if necessary, type the outcome of completing this step.	
5	Tab to the State column and, if necessary, type the name of the state into which the step moves the action.	
6	When you move out of the Action column, the next <i>new step</i> field displays underneath. Repeat steps 2 to 5 as many times as is necessary. The Structured Specification tab should now resemble the following illustration:	

Step	Action	See also
	<p>The screenshot shows the 'Structured Specification' tool interface. It features a toolbar with icons for save, undo, redo, and other actions. Below the toolbar is a table with columns: Step, Action, Uses, Results, and State. The table contains 8 rows of steps. Below this table is another tab labeled 'Entry Points' with columns: Step, Path Name, Type, and Join. This table contains 3 rows of entry points.</p>	

Notes




- By default, the steps begin with a user step (indicated by an actor icon) and alternate between user and system (indicated by a screen icon) steps. However, you can control the responsible entity assigned to a step by typing a keyword within the first 15 characters of the text in the Action column; either 'User' or 'Actor' for a user step, or 'System' for a system step. Once a step has been saved, you can change the responsible entity by either double-clicking on the icon or right-clicking on the step, and selecting the **Set Step as 'System'** or **Set Step as 'User'** context menu option as required
- An entry for the basic path displays in the Entry Points tab, as **Step 0** with no value in the **Join** column (the basic path does not rejoin itself)
- The values in the **Uses**, **Results** and **State** columns, whilst optional, are significant if you want to generate a diagram from the specification; if you type the name of an element linked to the current element (and listed in the Context References tab), the element name is highlighted and underlined
- In the **Uses** column, in edit mode, the **Manage Uses Context List** context menu option enables you to enable and disable **Uses** cells as linked element lists as well as text fields, across the whole model; when the feature is enabled, the column shows both context-linked elements and manually-entered text, and when disabled only text entries are listed (the context-linked entries are hidden and deactivated, but not deleted - they are re-activated when the feature is enabled)


- As you develop the scenario, you can move steps to different positions in the scenario, and insert new step lines within the body of the scenario
- You can also create constraints on the element that have an impact on the scenarios
- Repeat the **Scenario Steps** procedure for each scenario you have created; you can now adapt, enhance and interrelate the scenario specifications using the facilities of the Structured Specification tab:
 - Structured Specification Toolbar
 - Item context menu
 - Selected Text context menu
 - Entry Points context menu
 - Floating Toolbar








Learn more

- [Generate a Scenario from an Activity diagram](#) ^[873]
- [Scenarios](#) ^[852]
- [Structured Specification Item Context Menu](#) ^[859]
- [Generate Diagrams](#) ^[864]
- [Context References Tab](#) ^[874]
- [Structured Specification Toolbar](#) ^[857]
- [Scenario Constraints Tab](#) ^[875]
- [Structured Specification Entry Points Context Menu](#) ^[863]
- [Structured Specification Floating Toolbar](#) ^[864]
- [ScenarioStep Class](#) ^[2502]

The icons on the Structured Specification toolbar offer the following facilities:

Icon	Action	See also
	Save changes to the scenario specification.	
	Return to the basic path specification (if another specification is currently displayed).	
	(Only enabled when the basic path is displayed - you cannot add an alternate path to another alternate path or an exception path.) Create a branch from the selected step to an alternate path scenario - select the path from the displayed list. If the appropriate scenario does not yet exist, double-click on the <i>new path</i> line and type the scenario name, then click off the line and back on to it.	

Icon	Action	See also																								
	<p>Click on the OK button.</p> <ul style="list-style-type: none"> An entry for this alternate path displays in the Entry Points tab, as Step a of the basic path step it branches from; in the Join column, click on the drop-down arrow and select the number of the step at which action flows back to the basic path, or select End if the path terminates separately from the basic path You cannot add more than one branch - whether alternate or exception - to a step You cannot join the end of the branch - whether alternate or exception - to more than one step of the basic path. If you change the join point of the branch, any duplicates of that branch are updated to remain in synchrony with the join point <div data-bbox="368 949 1225 1189" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <table border="1"> <thead> <tr> <th colspan="4" style="background-color: #e0e0e0;">Entry Points Context References Constraints</th> </tr> <tr> <th>Step</th> <th>Path Name</th> <th>Type</th> <th>Join</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>In-House Account Processing</td> <td>Basic Path</td> <td>-</td> </tr> <tr> <td>2a</td> <td>Access1</td> <td>Alternate</td> <td>6</td> </tr> <tr> <td>4a</td> <td>ATM Path</td> <td>Alternate</td> <td>End</td> </tr> <tr> <td>6a</td> <td>Data Not Found</td> <td>Exception</td> <td>End</td> </tr> </tbody> </table> </div>	Entry Points Context References Constraints				Step	Path Name	Type	Join	0	In-House Account Processing	Basic Path	-	2a	Access1	Alternate	6	4a	ATM Path	Alternate	End	6a	Data Not Found	Exception	End	
Entry Points Context References Constraints																										
Step	Path Name	Type	Join																							
0	In-House Account Processing	Basic Path	-																							
2a	Access1	Alternate	6																							
4a	ATM Path	Alternate	End																							
6a	Data Not Found	Exception	End																							
	<p>(Only enabled when the basic path is displayed - you cannot add an exception path to another exception path or an alternate path.)</p> <p>Create a branch from the selected step to an exception path scenario - select the path from the displayed list. If the appropriate scenario does not yet exist, double-click on the <i>new path</i> line and type the scenario name, then click off the line and back on to it.</p> <p>Click on the OK button.</p> <ul style="list-style-type: none"> An entry for this exception path displays in the Entry Points tab, as Step a of the basic path step it branches from; in the Join column, click on the drop-down arrow and select the number of the step at which action flows back to the basic path, or select End if the path terminates separately from the basic path You cannot add more than one branch - whether alternate or exception - to a step You cannot join the end of the branch - whether alternate or exception - to more than one step of the basic path. If you change the join point of the branch, any duplicates of that branch are updated to remain in synchrony with the join point 																									

Icon	Action	See also
	Display the Manage Uses Context List dialog, which you use to set the Uses column entries to linked context reference lists.	Structured Specification Selected Text Context Menu ^[862]
	Move the currently-selected step one place up (including any <i>new step ...</i> entry).	
	Move the currently-selected step one place down (including any <i>new step ...</i> entry).	
	Display a list of diagrams that you can generate from the scenario; select the type of diagram that you want to generate.	Generate Diagram ^[864]
	Generate Test Cases based on this Use Case scenario; you can generate either internal Test Cases or External Test Cases.	Generate Test Cases ^[873]
	Delete the selected step from the scenario.	
	Display the Help topic for this tab.	

To display the context menu for a structured specification item, right-click on a step or blank line on the Structured Specification tab. The following options are available:

Option	Action	Shortcut	See also
Create Structure From Clipboard Text - New Line Delimited Create Structure From Clipboard Text - Sentence Delimited	Generate a set of steps from a text description or list captured on the clipboard; a new step is generated: <ul style="list-style-type: none"> • After each carriage return in the captured text (New Lines), or • For each sentence in the text; that is, after each full stop/space/capital letter combination (Sentences) <p>If a set of steps is already displayed, it is overwritten by the generated steps.</p>		
Create Structure From Generated Activity Diagram	Generate a set of steps from an Activity Diagram created for a Use Case. If a set of steps is already displayed, it is		Generate Scenario from Activity Diagram ^[873]

Option	Action	Shortcut	See also
	overwritten by the generated steps.		
Add Alternate Path	Create a branch from the selected basic path step to an alternate path.		Alternate path ^[857]
Add Exception Path	Create a branch from the selected basic path step to an exception path.		Exception path ^[858]
Insert Step Above	Insert a <i>new step...</i> line above the currently-selected step (press (Esc) to return this new line to the end of the scenario).	Shift + Insert	
Insert Step Below	Insert a <i>new step...</i> line below the currently-selected step (press (Esc) to return this new line to the end of the scenario).	Insert	
Insert End Step	Insert a <i>new step...</i> line at the end of the scenario.	Ctrl + N	
Set Step As 'User' Set Step As 'System'	Switch the entity responsible for performing the action of the selected step between user and system.		
Link Step to Use Case	<p>Either include the actions of an existing Use Case element, extend an existing Use Case element, or invoke a Use Case as the action of the selected step.</p> <p>Selecting the appropriate sub-option displays the Select Use Case dialog, which you use to browse for and select the required Use Case element.</p> <p>The appropriate <i>includes</i>, <i>extends</i> or <i>invokes</i> stereotyped connector is created between the current element and the selected Use Case.</p> <p>For the <i>include</i> and <i>extend</i> actions, any text in the Action field is overwritten by the link to the Use Case; for the <i>invoke</i> action, the following link is added to the end of the Action text:</p> <p>(<i>Invokes: <Use Case Name></i>)</p>		Select Use Case ^[880]
Merge With Step	<p>Merge the selected step with another.</p> <p>A list of the other steps in the scenario displays; click on the step to merge with the selected step.</p>		

Option	Action	Shortcut	See also
Move After Step	<p>Move the selected step to another position in the scenario.</p> <p>A list of the other steps in the scenario displays; click on the step after which to position the selected step.</p>		
Delete	<p>Delete the selected step; any subsequent steps are moved up one place.</p> <p>A prompt displays to confirm the deletion.</p>		

To display this context menu, *highlight* the text in a user-editable field within a step on the **Structured Specification** tab. The following options are available:

Option	Action	Shortcut	See also
Create	Create a glossary definition or a new element based on the highlighted text.		Glossary definition [977] New element [798]
Link Step to Use Case	<p>Either incorporate the actions of an existing Use Case element, extend an existing Use Case element or invoke a Use Case element, as the action of the selected step. Selecting the appropriate sub-option displays the Select Use Case dialog, which you use to browse for and select the required Use Case element.</p> <p>Any text in the Action field is overwritten by the link to the Use Case, except for the invoke action where the following link is added to the end of the Action text:</p> <p>(<i>Invokes: <Use Case Name></i>)</p>		Select Use Case [880]
Link to existing Element	<p>(Uses and Results fields only.) Create a Realization or Dependency relationship to a Requirement, Feature or other element elsewhere in the model.</p> <p>You select the element and connector types from submenu options, which then display the Select Element dialog, which you use to browse for and select the required element.</p>		Select Element [880]
Insert context reference	<p>Add a reference to an element stored elsewhere in the model, and create an entry for the element in the Context References tab.</p> <p>Selecting this option displays the Select Element dialog, which you use to browse for and select</p>		Context References [874] Select Element [880] Structured Specification

Option	Action	Shortcut	See also
	the required reference element.		Floating Toolbar ^[864]
Insert glossary definition	<p>Insert an existing glossary term at the cursor position. To select the term, double-click on it in the displayed list.</p> <p>When you select the term it is inserted into the field as highlighted and underlined text, which displays the definition when you move the cursor over it. If you highlighted part of the original text, the term overwrites that text.</p>		
Manage Uses Context List	<p>(On highlighted text or a specific cursor position in the Uses column.)</p> <p>Display the Manage Uses Context List dialog, which enables you to:</p> <ul style="list-style-type: none"> • Browse to or type in a context reference item • Re-sort the list with up and down arrow keys • Remove an entry from the list • Enable or disable the Uses Context List facility across the model <p>You can resize the dialog to show the full details of context reference items.</p> <p>If you enable the Uses Context List facility, you do not necessarily have to use the dialog to add context list items; you can type or insert an element name from the Context References tab in the Uses column, and that name is added to the Uses Context List.</p> <p>If you disable the Uses Context List facility, the context list items in the Uses column are hidden.</p>		Context References Tab ^[874]
Split Step	<p>Splits the selected step into two consecutive steps.</p> <p>The option is available only if you highlight a <i>portion</i> of the text in the selected field. The new step takes the highlighted text as its Action text.</p>		
Search for <text>	Displays a sub-menu of options for locating the selected text in a number of locations.		Code editor context menu ^[1805]

Option	Action	Shortcut	See also
Undo	Undo any unsaved changes you have just made in the step.		
Cut	Perform simple editing operations on the highlighted text.		
Copy			
Paste			
Delete			
Select All			

The Entry Points tab shows how the basic path, alternate path and exception path scenarios for the element are organized and interrelated. If an alternate path or exception path has been defined but has not yet been added to the basic path, it is not listed on this tab.

You can switch focus between the Entry Points tab and the Structured Specification tab by pressing (**Alt+Q**).

To display the context menu for this tab, highlight an entry and right-click on it. The following options are available:

Option	Action	Shortcut	See also
Edit Path	Display the steps of the scenario in the Structured Specification tab, with the first step highlighted.		
Join with Step	(Available only if the basic path scenario is displayed in the Structured Specification tab. Not available to edit the basic path scenario.) Highlight the Join field and its drop-down arrow. Click on the drop-down to define or change the step number at which the alternate or exception path rejoins the basic path. Select End if the path does not rejoin the basic path steps.		
Remove Entry Point	(Available only if the basic path scenario is displayed in the Structured Specification tab. Not available to delete the basic path.) Delete the relationship between the selected path and the basic path, and remove the entry from the Entry Points tab.		

Wherever a reference to another element exists on the Scenario tab (that is, where the text is highlighted and underlined), if you hover the cursor over the element name a short floating toolbar displays.

Use to

- Display the element Properties dialog
- Locate the element in its parent diagram
- Locate the element in the Project Browser


5.6.4.1.8.2 Generate Diagrams

If you have created a structured scenario, you can generate any of the following diagrams from that scenario:

- Activity
 - With ActivityParameter
 - With Action
 - With Action Pin
- Rule Flow
- State Machine
- Sequence
- Robustness

How to

To generate the required diagram

Icon	Step	Action	See also
	1	Create the scenario structure on the Structured Specification tab.	Structured Specification Tab ^[854]
	2	Click on the Generate Diagram icon in the toolbar on the tab.	
	3	<p>Click on the type of diagram to generate.</p> <p>Enterprise Architect generates the diagram and notifies you that generation is complete. Close the Properties dialog to review the diagram.</p> <p>If the diagram being generated already exists under the selected element, a prompt displays to overwrite or synchronize with that diagram.</p> <p>Select the appropriate radio button to:</p> <ul style="list-style-type: none"> • Overwrite the existing diagram (delete the existing diagram and elements, and create a new diagram and elements) or • Synchronize the elements in the existing diagram with the scenario steps (however, Sequence and Robustness 	

Icon	Step	Action	See also
		diagrams cannot be synchronized)	

Notes

- The **Synchronize elements in existing diagram** option enables the **Preserve Diagram Layout** checkbox, which you can select to preserve the existing arrangement of elements and connectors on the diagram; any *new* elements are added to the diagram in the default position, and you manually position them in the diagram as required; if you do not select the checkbox, the diagram is recast in the default layout

It is recommended that you uncheck the **Preserve Diagram Layout** checkbox if you are synchronizing elements with scenario steps:

- When new steps have been added or existing steps have been deleted or moved within the Use Case
- For the first time in a Use Case that has been imported from XML with the **Strip GUIDs** option selected
- For the first time in a Use Case that has been copied and pasted in the Project Browser, or
- For the first time in a Use Case whose containing package has been copied and pasted in the Project Browser

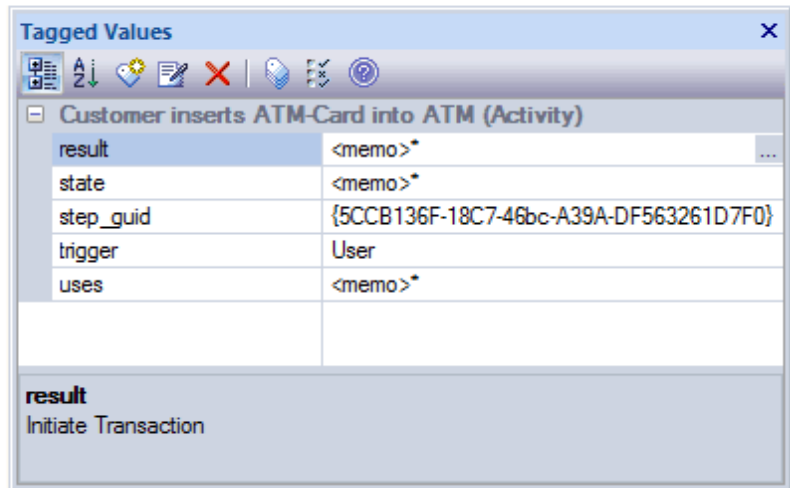
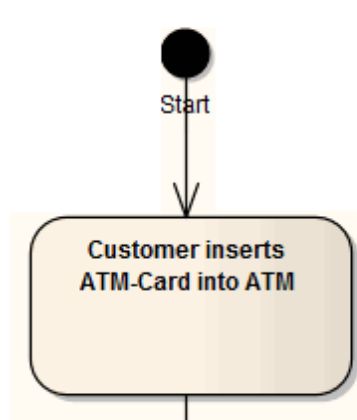
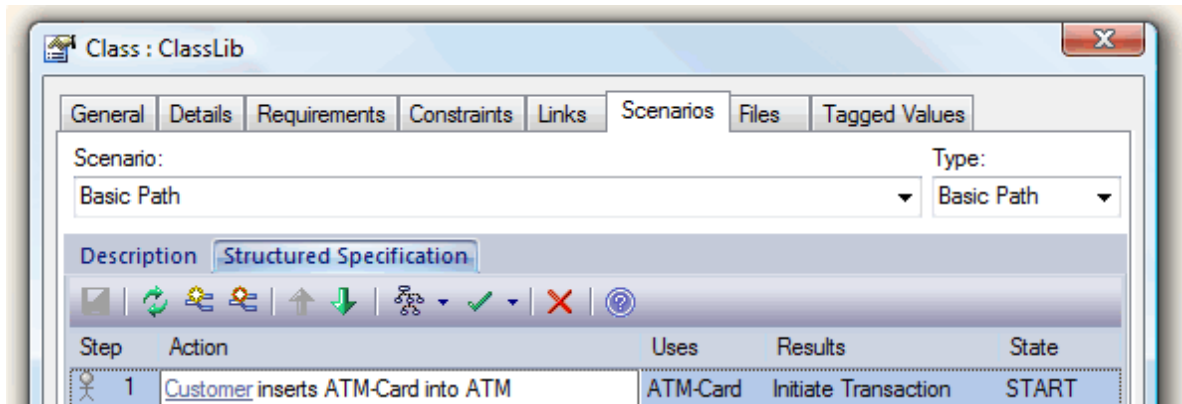
Learn more

- [Generate Activity Diagram](#) ^[865]
- [ActivityParameter](#) ^[865]
- [Action](#) ^[865]
- [Action Pin](#) ^[865]
- [Generate Rule Flow Diagram](#) ^[867]
- [Generate State Machine Diagram](#) ^[867]
- [Generate Sequence Diagram](#) ^[869]
- [Generate Robustness Diagram](#) ^[871]
- [Import from XML](#) ^[417]
- [Copy Element Between Packages](#) ^[808]
- [Copy A Package](#) ^[683]

An Activity is generated as a child of the selected element, to act as a container for the diagram.

- The scenario steps are modeled as Activities
- The values of the **Uses**, **Results** and **State** columns for each step are added as Tagged Values of the corresponding Activity

Example



Topics

Topic	Detail	See also
Activity with ActivityParameter	<ul style="list-style-type: none"> The values of the Uses and Results columns are modeled as ActivityParameters The value of the State column is added as a Tagged Value of the Activities ActivityParameters are added to the Project Browser and not to the diagram 	
Activity with Action	<ul style="list-style-type: none"> The scenario steps are modeled as Actions The values of the Uses, Results and State columns are added as Tagged Values of the Actions 	

Topic	Detail	See also
Activity with ActionPin	<ul style="list-style-type: none">• The scenario steps are modeled as Actions• The values of the Uses and Results columns are modeled as Input Pins and Output Pins respectively• The value of the State column is added as a Tagged Value of the Actions• ActionPins are added to the Project Browser and not to the diagram	

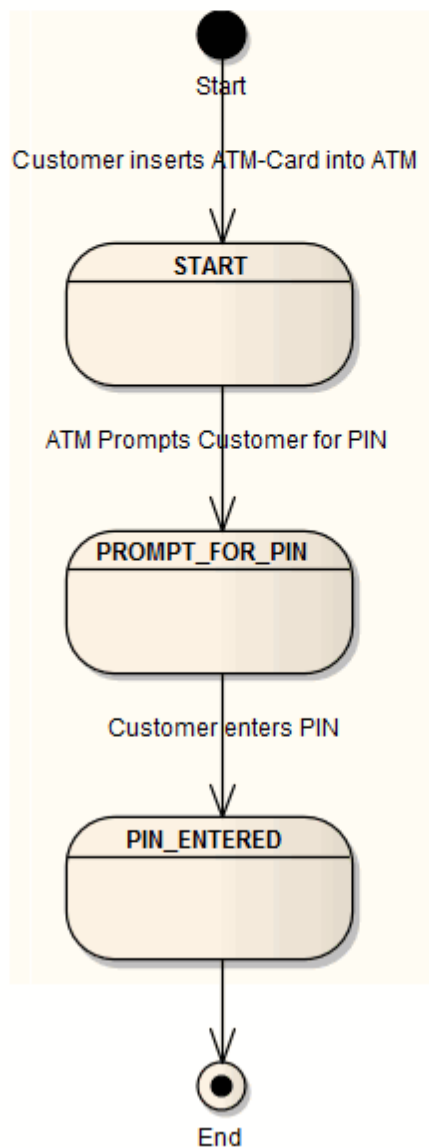
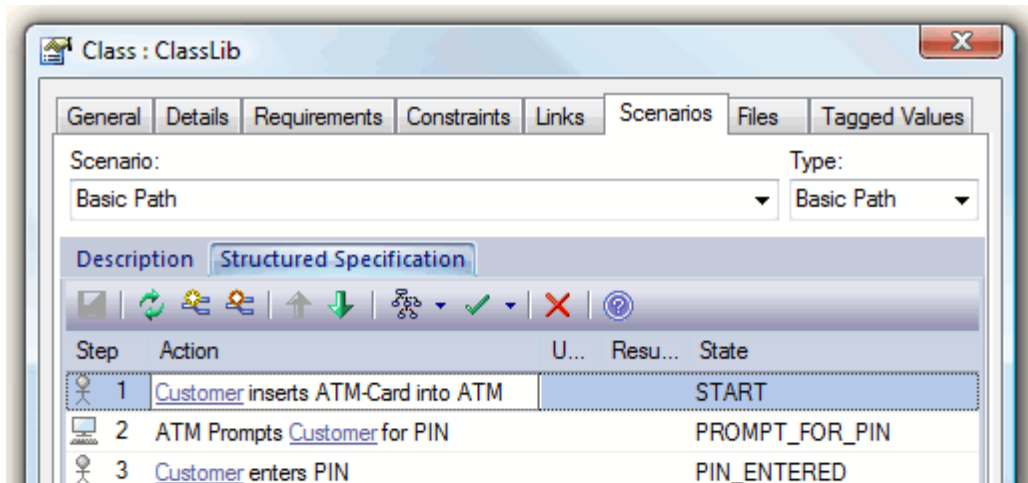
A Rule Flow Activity is created as a child of the selected element, to act as a container for the diagram.

- The scenario steps are modeled as RuleTasks.
- The values of the **Uses**, **Results** and **State** columns are added as Tagged Values of the RuleTasks

A StateMachine is created as a child of the selected element, to act as a container for the diagram.

- Each value in the *State* column is modeled as a State.
- The scenario steps become the Transition connectors between the States
- The values of the *Uses* and *Results* columns are added as Tagged Values of the Transitions

Example

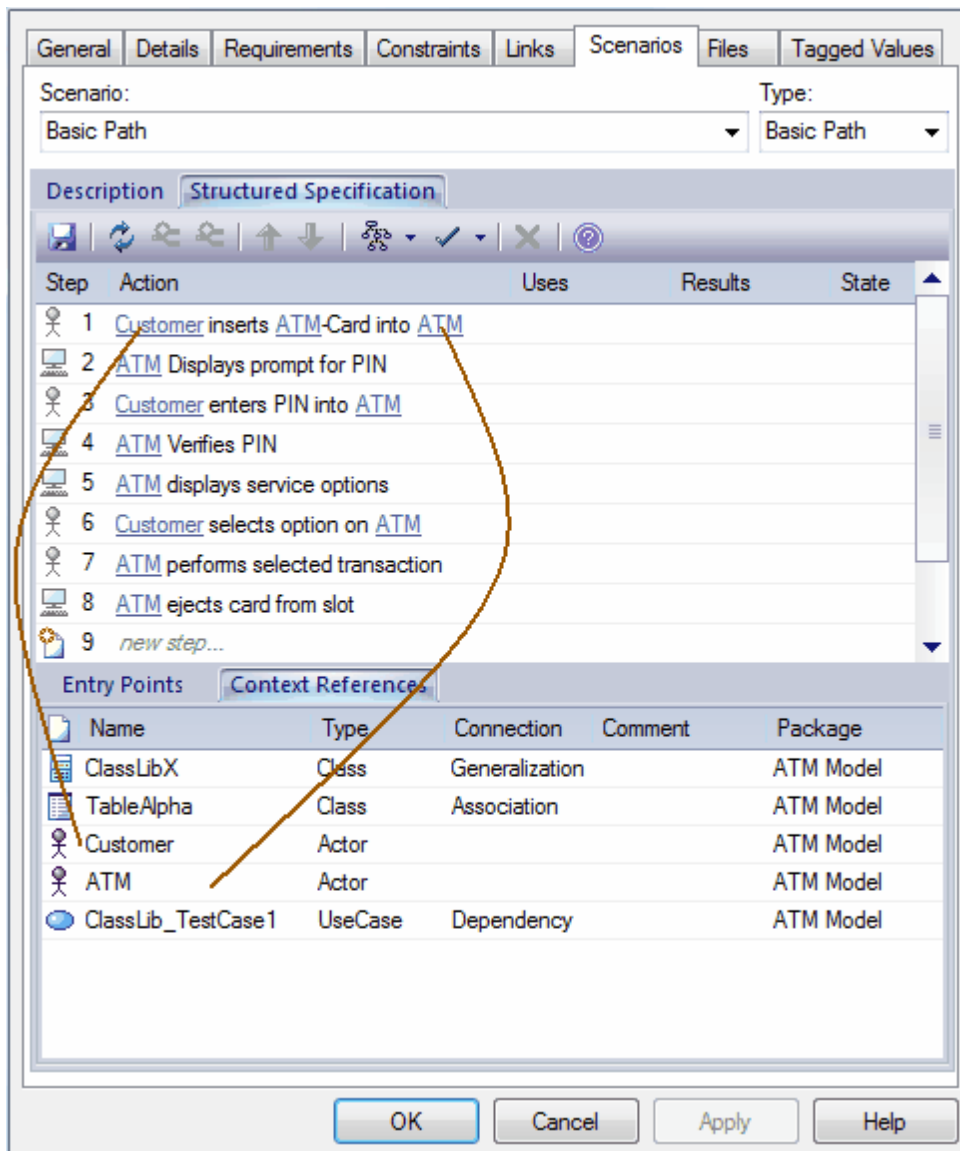


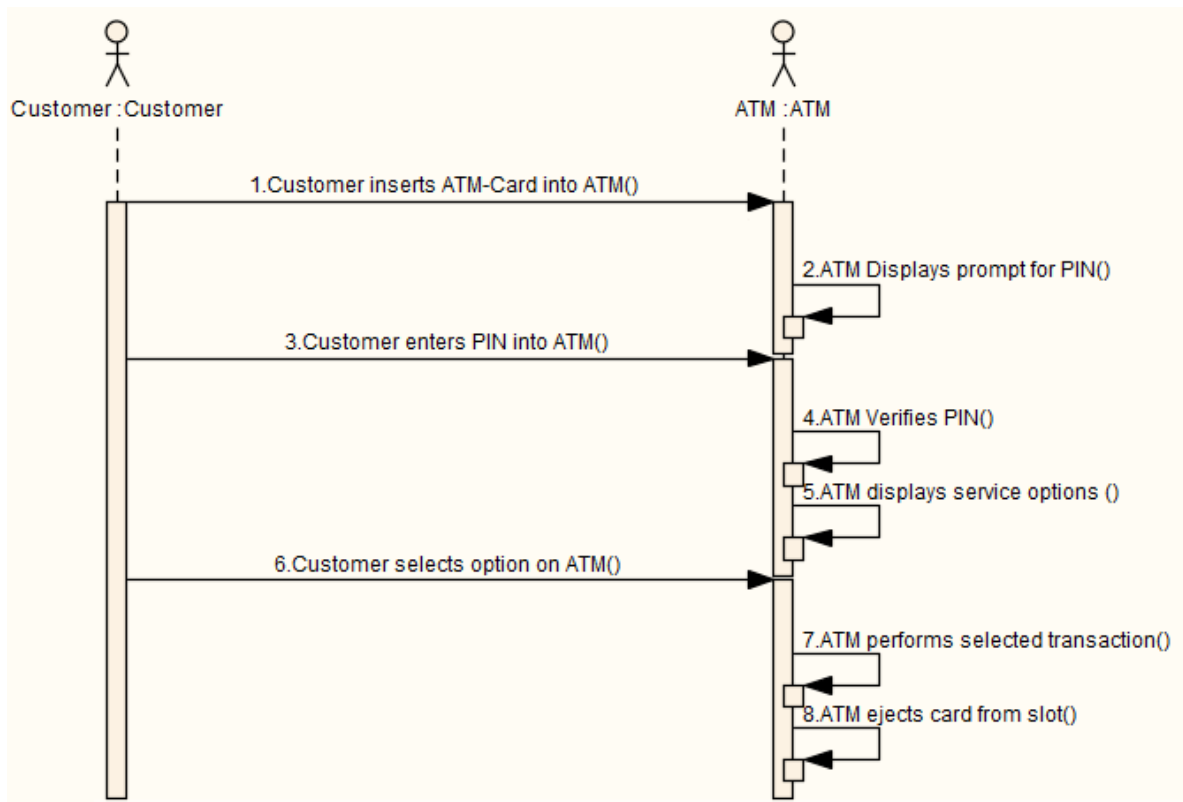
An Interaction is created as a child of the selected element, to act as a container for the diagrams - the Basic, Alternate and Exception paths are modeled as separate Sequence diagrams under the Interaction.

Topics

Topic	Detail	See also
Usage	<p>All the elements involved in the scenario should be identified in the Context Reference tab. That is, relationships must already exist between the scenario parent element and the other elements named in the scenario.</p> <ul style="list-style-type: none"> • Each Context Reference element named in a scenario step is modeled as a Lifeline • The step itself becomes the Message between an originator and its destination(s) • The first Context Reference element in a step is treated as the originator • The subsequent Context Reference element(s) become the destination(s) <p>Because the diagram generator acts on element names in the step, you should take care to avoid using the element names as normal text. For example, in step 1 in the dialog below, the term ATM-Card is interpreted as a reference to the ATM element, and two <i>Customer inserts ATM-Card into ATM</i> messages are generated for the step. (To avoid confusion, in the diagram the first Message has been deleted.)</p> <p>If the Show Sequence Numbering checkbox on the Diagram Sequence page of the Options dialog is not selected, the Message name on the Sequence diagram is prefixed by the step number, as shown in the diagram below. If the checkbox is selected, the message sequence number is shown instead of the step number.</p> <p>The values of the Uses, Results and State columns are added as Tagged Values of the Message.</p>	<p>Context Reference Tab ^[874]</p> <p>Sequence ^[555]</p>

Example

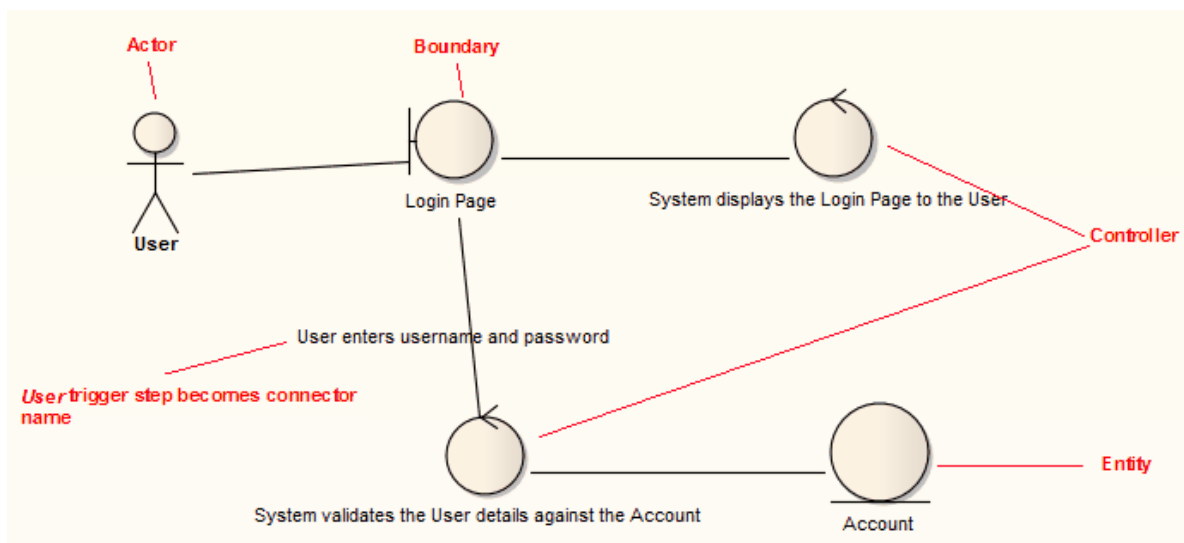




A Collaboration is created as a child of the selected element, to act as a container for the Robustness diagram.

Example

Step	Action
1	System displays the Login Page to the User
2	User enters username and password
3	System validates the User details against the Account



Notes

- All the elements involved in the scenario should be identified in the Context Reference tab; that is, relationships must already exist between the scenario parent element and the other elements named in the scenario
- Any values in the **Uses**, **Results** and **State** columns are ignored and not represented in the diagram
- Each UI element in a step becomes a Boundary element; a Dependency relationship is created from this Boundary element to the UI element (this connector is not shown on the diagram)
- Each Actor referenced in a step is dropped into the Robustness diagram as a simple link
- Each Class referenced in a step is dropped into the Robustness diagram as a simple link, and is given the stereotype *entity*
- Each step with a *System* trigger becomes a Controller; alternate/exception path Controllers are displayed with a red background color
- Each step with a *User* trigger becomes the name of the Association between Controllers

Learn more

- [Collaboration](#)^[1184]
- [Robustness Diagram](#)^[1508]
- [Context Reference Tab](#)^[874]
- [UI Elements](#)^[1657]
- [Boundary Elements](#)^[1663]
- [Dependency](#)^[1221]
- [Control](#)^[1665]
- [Associate](#)^[1210]

5.6.4.1.8.3 Generate Scenario From Activity Diagram

You can generate a range of diagrams from a scenario in an element. Conversely, you can also generate a structured scenario within an element from an Activity diagram, reverse engineering the steps from the diagram elements (effectively either regenerating the scenario within the Use Case, or transferring a scenario into another Use Case).

How to

To generate the scenario from the Activity diagram

Step	Action	See also
1	Open the element Properties dialog, select the Scenarios tab, and select the Structured Specification tab.	
2	Right-click in the empty space within the tab, and select the Create Structure From Generated Activity Diagram context menu option. The Select an Activity (generated from a UseCase Scenario) containing the Diagram dialog displays.	Select <Item> Dialog ^[880]
3	Search for and select the Activity containing the required diagram. Enterprise Architect validates the diagram (displaying the results in the Output window) and, if the diagram is valid, generates the scenario steps in the Structured Specification tab (replacing any existing scenario steps).	Output Window ^[164]

Notes

- The source Activity diagram must be generated from another Use Case Scenario
- Any existing scenario steps are deleted and replaced by the generated scenario
- This facility does not operate on the enhanced Activity diagrams generated from a Use Case - those generated with ActivityParameters, Actions and Action Pins

Learn more

- [Generate Diagrams](#)^[864]

5.6.4.1.8.4 Generate Test Cases

When you select a scenario and click on the **Test Case Generation** icon in the window toolbar, Enterprise Architect prompts you to select to generate either an Internal Test Case or an External Test Case.

Topics

Topic	Detail	See also
Internal Test	A test is generated for the basic path and each alternate and	Show Test Scripts

Topic	Detail	See also
Case	<p>exception path in the scenario, and added to the selected element.</p> <p>In addition, for each step in the basic, alternate and exception paths that has a value in the Results column, a test is generated and added to the selected element.</p> <p>A diagram is created under the selected element, to which the selected element and a Note (showing the element's Structured Specification) is added.</p> <p>The created tests are displayed in the <i>Test Scripts</i> compartment of the selected element on the diagram.</p> <p>Another way to view these tests is to click on the element and display the Testing window.</p> <p>These generated tests are written to the Scenario test tab of the Testing window; you can move the tests to another test-type tab if required.</p>	<p>compartment^[2229]</p> <p>The Testing Window^[2213]</p> <p>Move Or Copy Tests Between Categories^[2229]</p>
External Test Case	<p>A Test Case element is created, linked to the selected element using the Trace connector.</p> <p>A diagram is created under this Test Case element, to which the selected element, the Test Case element and a Note (showing the element's Structured Specification) is added.</p> <p>The created tests are displayed in the <i>Test Scripts</i> compartment of the Test Case element on the diagram.</p> <p>A test case is generated and added to the Test Case element for the basic path, and for each alternate and exception path.</p> <p>In addition, for each step in the basic, alternate and exception paths that has a value in the Results column, a test is generated and added to the Test Case element.</p> <p>You can review the tests within the Test Case element using the Testing window, as for Internal Test Cases.</p>	<p>Show Test Scripts compartment^[2229]</p> <p>Test Case^[1677]</p> <p>Trace^[1260]</p>

5.6.4.1.8.5 Context References Tab

On the Scenarios page of the element Properties dialog, the Context References tab displays a list of all elements that are either:

- Connected to the current element by any connector, on the current diagram or another, or
- Defined as a cross reference (or custom reference) on the current element

Topics

Topic	Detail	See also
Usage	<p>This page enables you to add custom references; right-click anywhere in the list and select the Add Reference context menu option.</p>	<p>Select <Item> Dialog^[880]</p>

Topic	Detail	See also
	<p>The Select Element dialog displays, in which you can locate and select the required cross reference element or elements.</p> <p>For each cross reference you can also use context menu options to delete the entry in the list, or to open the Comment field so that you can add or edit comment text.</p> <p>For each element in the Context References list, wherever the name of that element appears in the structured specification, the name is highlighted and underlined; press (Ctrl) + click on the highlighted name to view the element Properties dialog.</p>	

Learn more

- [Custom/Cross Reference](#)^[803]

5.6.4.1.8.6 Scenario Constraints Tab

The Structured Specification Constraints tab is a simple link to the Constraints tab of the element overall. It lists existing constraints; if you select the toolbar icons to add or edit a constraint, control switches to the element Constraints tab.

Learn more

- [Constraints](#)^[850]

5.6.4.1.9 Associated Files

An element can be linked to files external to the repository, using the Files page of the element's Properties dialog.

Field/Button	Usage	See also
File Path	Specifies the directory path and name of the file.	
Type	Display the local file or web address.	
Last Write	Display the date and time the file was last updated.	
Size	Display the size of the file.	
Notes	Indicates extra information about the file.	
Files	Display the list of linked files.	
Launch	Open the selected file.	

Field/Button	Usage	See also
	Local files open with their default application and web files open in the default browser.	

Notes

- Linked files are a good way to link elements to additional documentation and/or source code
- You can also insert hyperlinks in diagrams to other files, and launch them directly from the diagram

Learn more

- [Hyperlinks](#) ^[1668]

5.6.4.2 The Element Browser

The Element Browser window lists a range of added-on properties of the selected element.

Access [View](#) | [More Element Tools](#) | [Element Browser](#)

Topics

Topic	Detail	See also
Properties	<p>The following properties are listed, where they are present in the element:</p> <ul style="list-style-type: none"> • Operations • Attributes • Linked Features (Linked Operations and Attributes) • Tagged Values • Constraints • Internal Requirements (Responsibilities) • Attached Files • Relationships • Scenarios • Maintenance Items • Testing Items • Project Management Items 	<p>Operations ^[900]</p> <p>Attributes ^[885]</p> <p>Linked Features (Linked Operations and Attributes) ^[944]</p> <p>Tagged Values ^[969]</p> <p>Constraints ^[850]</p> <p>Internal Requirements (Responsibilities) ^[848]</p> <p>Attached Files ^[875]</p> <p>Relationships ^[851]</p> <p>Scenarios ^[852]</p> <p>Maintenance Items ^[2237]</p> <p>Testing Items ^[2213]</p> <p>Project Management Items ^[447]</p>

Notes

- If you double-click on the element name at the top of the dialog (the root node) the Properties dialog for the element displays, at the General tab
- If you right-click on a folder name, the **Edit <object>** context menu option displays; when you select this option, the appropriate window or dialog displays, or the element Properties dialog displays the appropriate tab, to enable you to update the items in the folder
- If you double-click on an item within the *Requirements*, *Scenarios* or *Constraints* folders, the docked window or view displays with the focus on the selected item
- If you double-click on an item within the *Relationships* folder, the Properties dialog displays for that relationship
- You can drag a relationship from the Element Browser onto a diagram; if the element at one or both ends of the relationship is not already on the diagram, the element is also added to the diagram, and if the relationship is already on the diagram, but hidden, it becomes visible again
- Instead of dragging the relationship onto the diagram, you can right-click on it and select context menu options to place the relationship and elements on the diagram, and to show or hide the relationship
- The relationship context menu also provides options for locating the related element in the Project Browser, for displaying the Properties dialog of the related element, and locating the related element in all the other diagrams in which it is used
- If you double-click on an item within the *Files* folder, the file opens either on a separate tab in the Diagram View workspace (if the file can be opened within Enterprise Architect) or in the default Windows viewer/editor for the file type (if the file cannot be opened within Enterprise Architect)
- If you double-click on an operation or attribute in the *Operations* or *Attributes* folders, the appropriate Properties dialog displays
- If you right-click on an attribute or operation, whether in the *Attributes* or *Operations* folder or in the *Linked Features* folder, a context menu displays that enables you to display the source code; alternatively, click on the attribute or operation and press (**Ctrl+E**), or - in the *Linked Features* folder - double-click on the attribute or operation
- You can drag an Operation or Attribute from the Element Browser to another element on a diagram

- The source code viewer in which the source code displays depends on which editor you select as the default, either for the project as a whole or for a specific programming language; if you select the Enterprise Architect internal editor, the code displays in the Source Code Viewer with the cursor positioned on the selected feature
- The right-click context menu for an operation also provides options to set a number of types of recording marker

Learn more

- [Scenarios and Requirements Window](#)^[879]
- [Connector Properties](#)^[961]
- [Options - Code Editors](#)^[1895]
- [Editing Source Code](#)^[1799]
- [Recording Markers](#)^[2154]

5.6.4.3 Properties Window

The Properties window provides a convenient way to view (and in some cases edit) common properties of elements. When an element is selected, the Properties window shows the element's name, stereotype, version, author, dates and other pertinent information.

Access [View | Element Properties](#)

Topics

Topic	Detail	See also
Properties Sections	The Properties window is divided into three expandable sections: <ul style="list-style-type: none"> • <Element type> Settings - for the basic element details • Project - for general housekeeping settings • Advanced - only active for generalizable elements 	

Notes

- When you click on a field name, a brief explanation of that field displays at the bottom of the Properties window, unless you have selected the **Hide Properties Info Section** checkbox on the General page of the Options dialog
- If you click on the field value for an editable field, a drop-down arrow displays that enables you to select a different value; one of the values offered might display a more advanced list or a dialog for adding new objects or selecting multiple objects, as the **Stereotype** field does

Learn more

- [Hide Properties Info Section](#) ^[545]
- [Apply Stereotypes](#) ^[1265]


5.6.4.4 The Scenarios & Requirements Window

The Scenarios & Requirements window provides a convenient way to quickly add, view, edit and delete rules applied to an element. The window shows details of the entities that impose such rules or restrictions on the element, namely:

- The element's internal responsibilities or requirements
- The element's internal constraints (not external Constraint notes)
- The element's scenarios


Access [View | More Element Tools | Scenarios & Requirements](#) (Ctrl + Shift + 3)

Topics

Topic	Detail	See also
Usage	<p>The Scenarios & Requirements window is typically used to examine Use Case and Test Case elements, and any other elements that realize an external Requirement.</p> <p>For convenience, you can display the window as either a dockable window around the edge of your workspace, or as a view in the center of your workspace (in which the Scenarios, Constraints and Requirements are shown on separate tabs).</p> <p>Use the  icon in the window toolbar to toggle between these display options.</p>	Notes ^[1147]

How to

To review an internal requirement, constraint or scenario for an element

Step	Action	See also
1	<p>Select the element in the Project Browser or diagram and select the Scenarios & Requirements menu option.</p> <p>The Scenarios & Requirements window displays.</p>	
2	<p>Either:</p> <ul style="list-style-type: none"> • Click on the required item in the list panel on the left of the window, or • If the Element Browser is not already displayed, click on the  icon and select the required item from the <i>Constraints</i>, <i>Requirements</i> or <i>Scenarios</i> folder 	Element Browser ^[876]

Step	Action	See also
	The appropriate screen displays, with the details of the selected item in the fields.	
3	<p>The toolbar icons, from left to right, enable you to:</p> <ul style="list-style-type: none"> • Add a new rule to the current screen (click on the appropriate folder to create an item of a different type) • Save the new or edited item • Save the current (edited) item as a new item • Delete the current item • Sort the contents of the selected folder into alphabetical order • (Print icon unavailable) • Display the Element Browser window, or highlight the current item in the Element Browser window • Switch between a docked window display and a workspace view display <p>You can also add or edit formatted notes in the Notes field, using the Notes toolbar at the top of the field.</p> <p>The list panel also provides a context menu that provides options for adding a new item to a folder, deleting the selected item or, for a requirement, converting an internal responsibility into an external Requirement element.</p>	<p>Notes Toolbar ^[979]</p> <p>Converting an Internal Responsibility Into an External Requirement ^[1486]</p> <p>Scenarios ^[852]</p> <p>Constraints ^[850]</p> <p>Requirements ^[848]</p>

5.6.4.5 Select <Item> Dialog

The Select <Item> dialog is a multi-purpose browser and search tool for locating model items such as Classifier elements, properties, attributes and behaviors. The <Item> in the dialog title changes to represent the type of item the original operation is working on.

Topics

Topic	Detail	See also
Usage	<p>The dialog is called in a range of operations; for example, setting:</p> <ul style="list-style-type: none"> • The base type or classifier for an Object, Swimlane or Lifeline • Classifiers for the return types for operations • Classifiers for generalization sets • The Information Item or classifier elements conveyed or realized on an Information Flow • The associated behavior for a behavior call 	<p>Using Classifier ^[899]</p> <p>Objects ^[1196]</p> <p>Swimlanes ^[751]</p> <p>Lifeline ^[1144]</p> <p>Using Information Flows ^[226]</p> <p>Return Types ^[901]</p> <p>Generalization Sets ^[955]</p>

	<ul style="list-style-type: none"> • The type and return type for operation parameters • Activities for State transitions • Use Case Scenarios • Pattern element defaults • Activities from which to generate code • The values of Tagged Values 	Associate with Different Behaviour ^[91] Insert Operation Parameters ^[92] Transitions ^[1260] Generate Scenarios from Activity Diagram ^[873] Use a Pattern ^[1274] Assigning Tagged Values ^[972]
Multiple Selection	<p>Where an operation permits the selection of multiple items, the Select <Item> dialog is automatically enabled to support this; to select the items, press (Ctrl) as you click on each item.</p> <p>If you want to cancel a selected item, press (Ctrl) and click on the item.</p> <p>Having selected an item, you can continue to expand and browse the hierarchy, and/or search for items; the dialog retains the existing selections until you click on the OK button.</p>	

How to

To select a required item

Step	Action	See also
1	<p>During an operation, when it is necessary to locate an element or feature, click on the (...) (browse) button.</p> <p>The Select <Item> dialog displays.</p>	
2	<p>If required, in the Go To Namespace field select a namespace to reduce the scope of the displayed hierarchy.</p> <p>The dialog opens the section of the hierarchy associated with that namespace, and closes all previously-open sections associated with other namespaces.</p>	
3	<p>You can either:</p> <ul style="list-style-type: none"> • Expand the selected area of the hierarchy on the Browse tab, or any other package, and locate the required item (go to step 5) or • Click on the Search tab and, in the Find field, type a partial or complete text string to search for the item 	

Step	Action	See also
4	<p>On the Search tab, you can filter the search further by selecting the Whole Word and Match Case checkboxes.</p> <p>Each list entry shows the name of the item, the type, any stereotype the item has, the immediate package in which the item is held, and any successive parent packages (the package path).</p> <p>You can either:</p> <ul style="list-style-type: none"> • Select the item immediately on the Search tab or • Right-click on one item or a group of items and select the Locate item(s) in tree context menu option; this redisplay the Browse tab and highlights each selected item in the <namespace> hierarchy 	
5	Click on the required item.	
6	Click on the OK button.	

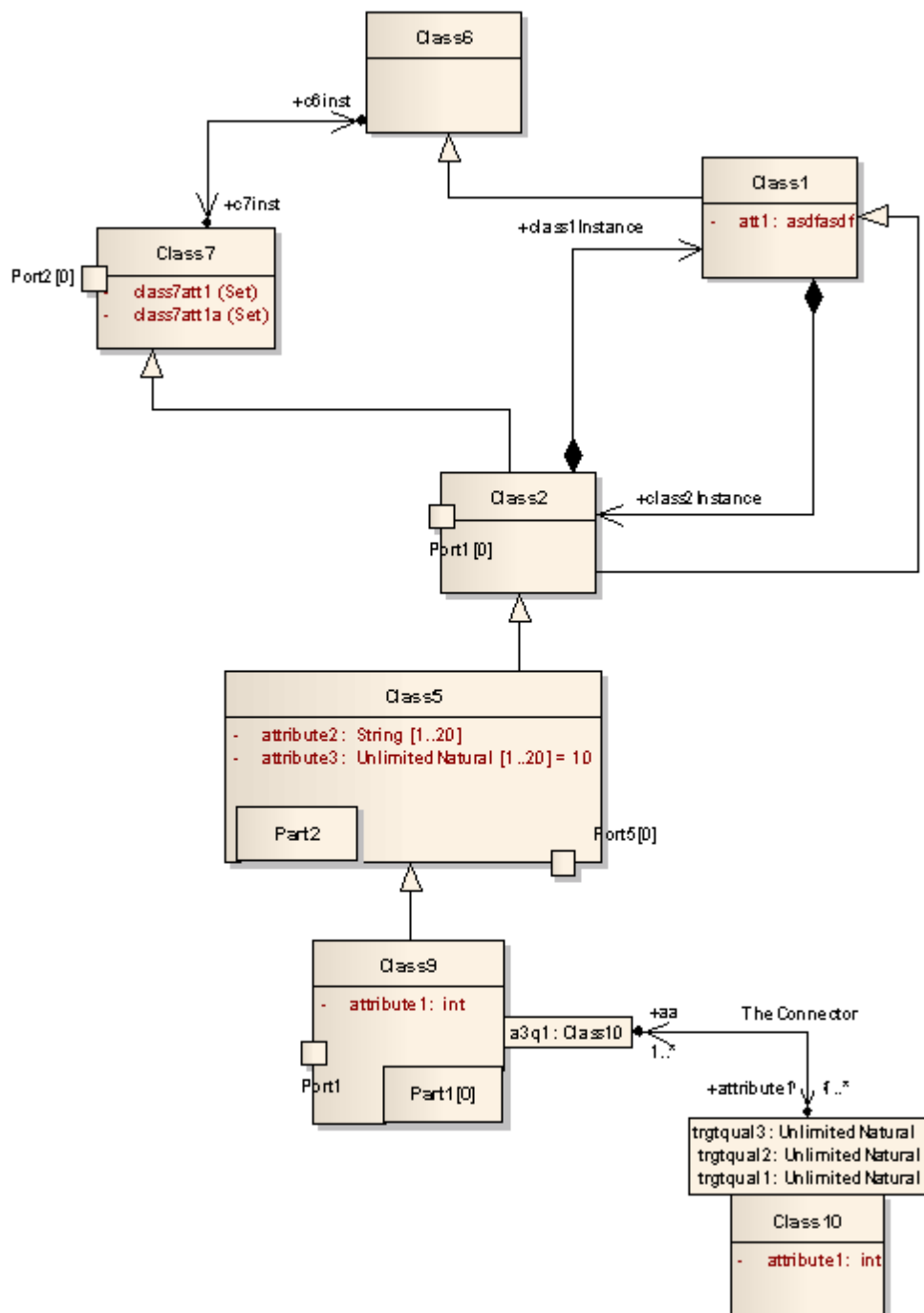
Notes

- When you have selected an item, the Select <Item> dialog retains the context and item; next time you display the dialog, if the context is similar the dialog opens to the same Namespace and item - for example, if you have selected an activity for a State transition and you start to do the same for another transition, the dialog opens to the activity you previously selected
- If the context is totally dissimilar, the dialog opens with the Namespace <any> and a collapsed model hierarchy
- If the available items do not meet your requirements, you can create a new item and define the appropriate properties - click on the **Add New** button; the appropriate Add <Item> dialog displays, on which you define the required item
- The **Add New** button is not always available, depending on the context and the type of item being searched for

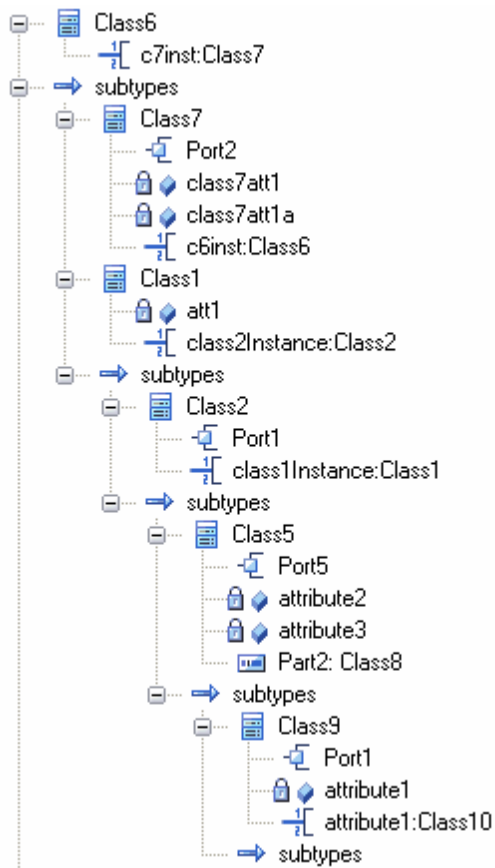
5.6.4.5.1 Select Property Dialog

The Select Property dialog is a specific instance of the Select <Item> dialog; it is used to select Ports, Parts, Attributes and Association Ends as *redefined* or *subsetting* properties, from a *hierarchical list* of Classes and their contained properties in the model. For example, consider the section of the model below:

Example



This would be represented in the dialog as follows:



Locate and click on the required object to select it, then click on the **OK** button (to select several objects at once, press and hold (**Ctrl**) while you click on each object).

Notes

- AssociationEnds should be owned by the Class to be listed in the dialog

Learn more

- [Select <Item> Dialog](#)^[880]

5.6.4.5.2 Set Feature Dialog

The Set Feature dialog is the Set Operation dialog used to change the operation represented by an Action on an Activity diagram.

As the Set Operation or Set Attribute dialog, it is also used to set the *Value* operation or attribute for Tagged Values of type RefGUID or for the target of a hyperlink in a diagram.

How to

To locate and select a value operation or attribute

Step	Action	See also
1	The Set Operation (or Set Attribute) dialog displays, with the model hierarchy opened at the point at which you selected the original operation or attribute.	
2	If required, in the In Namespace field, click on the drop-down arrow and select another model that contains the required operation or attribute. The package hierarchy for that model displays.	
3	Browse through the hierarchy, or use the Search tab to locate the required operation or attribute, then double-click on the item to select it.	Select <Item> Dialog ^[880]

Learn more

- [Class Operations In Activity Diagrams](#)^[1100]
- [Pre-defined Structure Types - RefGUID](#)^[1388]
- [Hyperlinks](#)^[1669]

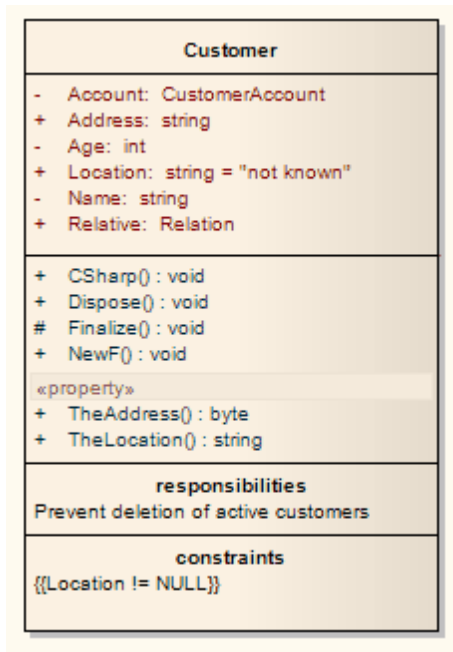
5.6.5 Attributes

Attributes are features of an element that represent the properties or internal data elements of that element. Not all element types support attributes, and others have restrictions - for example, attributes of Interfaces must have Public scope.


Elements with attributes (typically Classes) display their features in diagrams in the manner illustrated below. Attributes display in the first compartment of properties in colored text - the default color is red (for example, **Age : int**).

Example

For a Customer Class, **CustomerName** and **CustomerAddress** can be attributes. Attributes have several important characteristics, such as **type**, **scope** (visibility), **static**, **derived** and **notes**.



Topics

Topic	Detail	See also
Create and Modify Element Attributes	<p>In the Project Browser, Classes with attributes have their features collected beneath them, each preceded by a blue box ().</p> <p>In the Diagram view or Project Browser, either:</p> <ul style="list-style-type: none"> • Right-click on the element to be edited, and from the context menu select the Attributes menu option, or • Click on the element and press (F9) <p>The <Element name> Attributes dialog displays.</p> <p>You can also copy or move existing attributes between elements.</p> <p>These facilities are only available if the target element supports attributes.</p>	<p>Attributes Dialog - General ⁸⁸⁷</p> <p>Copy Features Between Elements ⁸²³</p> <p>Move Features Between Elements ⁸²⁴</p>

Notes

- If you make changes and do not save them, the **Cancel** button prompts you to save or cancel the changes, whilst the **Close** button closes the dialog immediately and does not save the changes
- If you are creating many attributes, go to the Attribute/Operations page of the Options dialog (**Tools | Options | Source Code Engineering | Attribute/Operations**) and select the **After save, re-select edited item** checkbox; now, when you create an attribute and click on the **Save** button, the dialog fields clear ready for you to enter the details of the next attribute - this helps you when you want to create attributes quickly and might not necessarily want to fully define each one as you create it

- If the parent element provides source or target roles for a connector, the connector can be attached to a specific attribute (feature)

Learn more

- [Attributes Dialog - Detail](#) ^[889]
- [Attributes Dialog - Constraints](#) ^[890]
- [Attribute Tagged Values](#) ^[891]
- [Create Properties](#) ^[892]
- [Display Inherited Attributes](#) ^[893]
- [Create Object From Attribute](#) ^[896]
- [Connect to Element Feature](#) ^[944]

5.6.5.1 Attributes Dialog - General

This topic describes the fields and options of the General page of the Attributes dialog.

Reference

Field	Usage	See also
Name	Display the name of the attribute; for a new attribute, type the name (with no spaces).	
Type	<p>Display the attribute's type; if necessary, either:</p> <ul style="list-style-type: none"> • Click on the drop-down arrow and select a different type, or • Click on the (...) button to open the Select <Item> dialog and select or define a different attribute classifier type that might not be in the Type drop-down list <p>The type can be defined by the code language (data type) or by a classifier element; when you click on the drop-down arrow, the first set of values in the list provides the data types, and the second (longer) set provides the possible classifiers.</p> <p>To add new code language data types that can be displayed in this list, see the <i>Data Types</i> topic.</p>	Data Types ^[987] Instance Classifier ^[880]
Initial Value	Display an optional initial value; if necessary, type in or browse for a new initial value.	
Stereotype	<p>Define the optional stereotype of the attribute; if necessary, either:</p> <ul style="list-style-type: none"> • Type a different stereotype name • Click on the drop-down arrow and select a stereotype, or • Click on the (...) button and browse for the stereotype 	

Field	Usage	See also
Alias	Display an optional alias for the attribute; if necessary, type in a new alias.	
Scope	Define the attribute as Public , Protected , Private or Package ; if necessary, click on the drop-down arrow and select a different scope.	
Static	Indicate that the attribute is a static member.	
Const	Indicate that the attribute is a constant.	
Is Literal	(For Enumeration elements) Defaults to selected, to define the attribute as an enumeration literal. Deselect the checkbox to define the attribute as a normal element attribute. In the <i>Attributes</i> compartment on the diagram, the enumeration literals are listed separately, above the normal attributes (ensure that the Stereotype field for the normal attribute is not set to enum).	
Containment	Define the containment type (By Reference , By Value or Not Specified); if necessary, click on the drop-down arrow and select a different containment type.	
Property	Indicate that the attribute has automatic property creation; when you select this checkbox, the Create Property Implementation dialog displays.	Create Properties [892]
Notes	Enter any free text notes associated with the attribute; you can format the notes text using the Notes toolbar at the top of the field.	Notes Toolbar [979]

Topics

Topic	Detail	See also
Usage	To change the position of a selected attribute in the list at the bottom of the dialog, click on the Scroll Up or Scroll Down (hand) buttons. To review an existing attribute, click on the attribute name in the list. To delete an existing attribute, click on the attribute name in the list and click on the Delete button. To create a new attribute, either:	

Topic	Detail	See also
	<ul style="list-style-type: none"> Click on the New button to clear the fields for new information, or Click on an existing attribute name in the list, and click on the Copy button; a prompt displays for you to enter the name of the copy <p>If you have changed the attribute details, click on the Save button to save the changes.</p>	

Notes

- By default, the attributes are listed in alphabetical order; before changing this sequence, you must deselect the **Sort Features Alphabetically** checkbox on the Objects page of the Options dialog (**Tools | Options | Objects**)

5.6.5.2 Attributes Dialog - Detail

This topic describes the fields and options of the Detail page of the Attributes dialog.

Reference

Field	Action	See also
Multiplicity		
Lower bound	Define a lower limit to the number of elements allowed in the collection.	
Upper bound	Define an upper limit to the number of elements allowed in the collection.	
Allow Duplicates	Indicate that duplicates are allowed. This field maps to the UML property <i>isUnique</i> , with a value FALSE .	
Ordered Multiplicity	Indicate that the collection is ordered.	
Redefined Property	Review the redefined properties for the attribute. You can add redefined properties by clicking on the Add button	Select Property

Field	Action	See also
	to display the Select Property dialog.	dialog ^[882]
Subsetted Property	Review the subsetted properties for the attribute. You can add subsetted properties by clicking on the Add button to display the Select Property dialog.	
Collection		
Attribute is a Collection	Indicate that the attribute is a collection (array), so that it can contain multiple concurrent values rather than a single value.	
Container Type	Enter the name of the container type.	
Other		
Transient	(For Java code) indicate that the attribute can change regardless of what the code is performing.	
Derived	Indicate that the attribute is derived from one or more other attributes; for example, full name can be derived from the <i>last name</i> and <i>first name</i> attributes.	
isID	When selected (true), enables the attribute to be used to uniquely identify an instance of the containing Class. Defaults to unselected (false).	
Qualifiers	Click on this button to add Qualifiers to the attribute. The Qualifiers dialog displays.	Qualifiers ^[1211] Qualifiers dialog ^[1212]
Save	Click on this button to save changes to this page.	

5.6.5.3 Attributes Dialog - Constraints

Attributes can have constraints associated with them; typically these indicate such things as maximum value, minimum value and length of field.

These constraints are managed on the Constraints page of the Attributes dialog.

Topics

Topic	Detail	See also
Usage	<p>Select the Constraints page to define the constraints.</p> <ul style="list-style-type: none"> To review an existing constraint, click on the constraint name in the panel at the bottom of the dialog To delete an existing constraint, click on the constraint name in the panel and click on the Delete button To create a new constraint, click on the New button <p>Review, edit or complete the fields as indicated in the table below.</p> <p>If you have created or edited the data, click on the Save button to save the changes.</p>	

Reference


Field	Usage	See also
Constraint	Type in the constraint name.	
Type	Type in the constraint type.	
(Notes)	Type in any comments or notes concerning the constraint.	

5.6.5.4 Attribute Tagged Values

You can define Tagged Values for an attribute. Tagged Values are a convenient means of extending the properties a model element supports. This in turn can be used by code generators and other utilities to transform UML models into other forms.

How to

To add a Tagged Value to an attribute

Step	Action	See also
1	Click on the Tagged Values tab of the Attribute Properties dialog.	
2	Click on the New tag button (). The Tagged Value dialog displays.	
3	In the Tag field, type the tag name or click on the drop-down arrow and select a defined	

Step	Action	See also
	tag.	
4	If appropriate, in the Value field type a specific value for the tag.	
5	Click on the OK button to confirm the operation. The tag name and value are displayed under the attribute in the Tagged Values tab.	

Notes

- Tagged Values are supported for attributes, operations, objects and connectors
- You can define custom tags by creating a Custom Tagged Value Type


Learn more

- [Creating a Custom Tagged Value Type](#) 

5.6.5.5 Create Properties

Enterprise Architect has capabilities for automatically creating properties in various languages. Property creation is controlled from the General page of the Attributes dialog.

Topics

Topic	Detail	See also
Usage	<p>On the General page, select the Property checkbox.</p> <p>The Create Property Implementation dialog immediately displays.</p> <p>The Language panel defaults to the Class language; however, you can change this and generate the properties for any language, including Action Script.</p> <p>Each language has slightly different syntax and generates slightly different results</p> <ul style="list-style-type: none"> • Java and C++ generate get and set functions • C# and VB.Net create property functions • Delphi creates get and set functions as well as a specialized Delphi property Tagged Value <p>Type in the required details and click on the OK button.</p> <p>Enterprise Architect generates the required operations and properties to comply with the selected language.</p> <p>Note that get and set functions are stereotypes with «<i>property</i></p>	<p>Compartments </p>

Topic	Detail	See also
	<p><i>get</i>» «<i>property set</i>» making it easy to recognize property functions; you can also hide these specialized functions by deselecting the Property Methods checkbox in the Features tab of the Diagram Properties dialog for a specific diagram (select the Diagram Properties menu option). This makes it easier to view a Class, uncluttered by many <i>get</i> and <i>set</i> methods.</p> <p>Note that for Delphi you must enable the Tagged Values compartment to see the generated properties.</p>	

5.6.5.6 C# Automatic Property Support

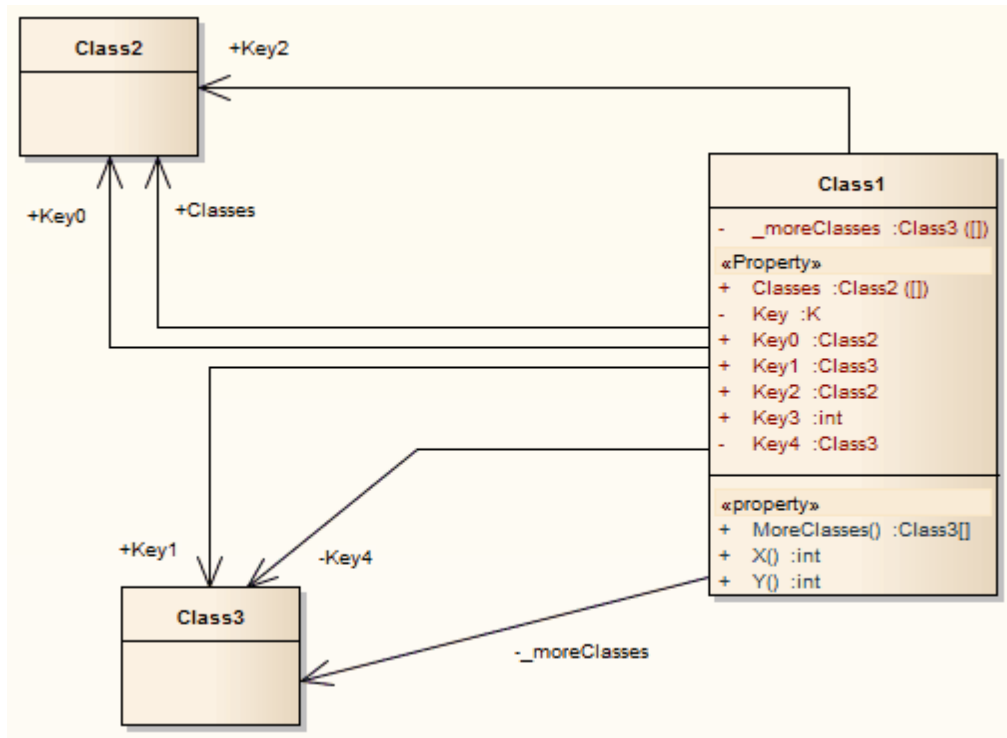
For C# at and beyond release 4.0, Enterprise Architect enables you to model *automatic properties* as «*Property*» stereotyped attributes.

Topics

Topic	Detail	See also
Properties of Attributes	<p>When you create the attribute, on the General page of the Attributes dialog simply set the Stereotype field to Property; the generated code resembles the following:</p> <pre>public class Class1 { public Class2() Classes {get; set;} }</pre> <p>This is in contrast to the Enterprise Architect facilities for modeling <i>get/set</i> methods as properties of an attribute in various other languages, generating code such as:</p> <pre>public class Class1 { public Class3() m_delivery; : public Class3() delivery { get{ } set{ } } }</pre> <p>You cannot create properties with <i>get/set</i> methods on an automatic properties attribute; whilst the «<i>Property</i>» stereotype is assigned in the Attributes dialog, the Property checkbox is grayed out.</p>	Create Properties ^[892]

Topic	Detail	See also
Tagged Values	<p>You can also add the following Tagged Values to automatic properties to denote specific characteristics, if required:</p> <ul style="list-style-type: none"> • writeonly - the property only has a <i>set method</i> <p style="padding-left: 40px;">For example: <code>public int Prop1 { set; }</code></p> <ul style="list-style-type: none"> • readonly - the property only has a <i>get method</i> <p style="padding-left: 40px;">For example: <code>public int Prop2 { get; }</code></p> <ul style="list-style-type: none"> • GetScope - specifies the property's <i>Get Scope</i> <p style="padding-left: 40px;">For example: <code>public int Prop3 { protected get; set; }</code></p> <ul style="list-style-type: none"> • SetScope - specifies the property's <i>Set Scope</i> <p style="padding-left: 40px;">For example: <code>public int Prop4 { get; protected set; }</code></p>	
Reverse Engineering	<p>Imported automatic properties generate attributes with a «<i>property</i>» stereotype and appropriate Tagged Values, and - if the property type is a recognized element in the model - an Association connector.</p>	

Example



5.6.5.7 Display Inherited Attributes

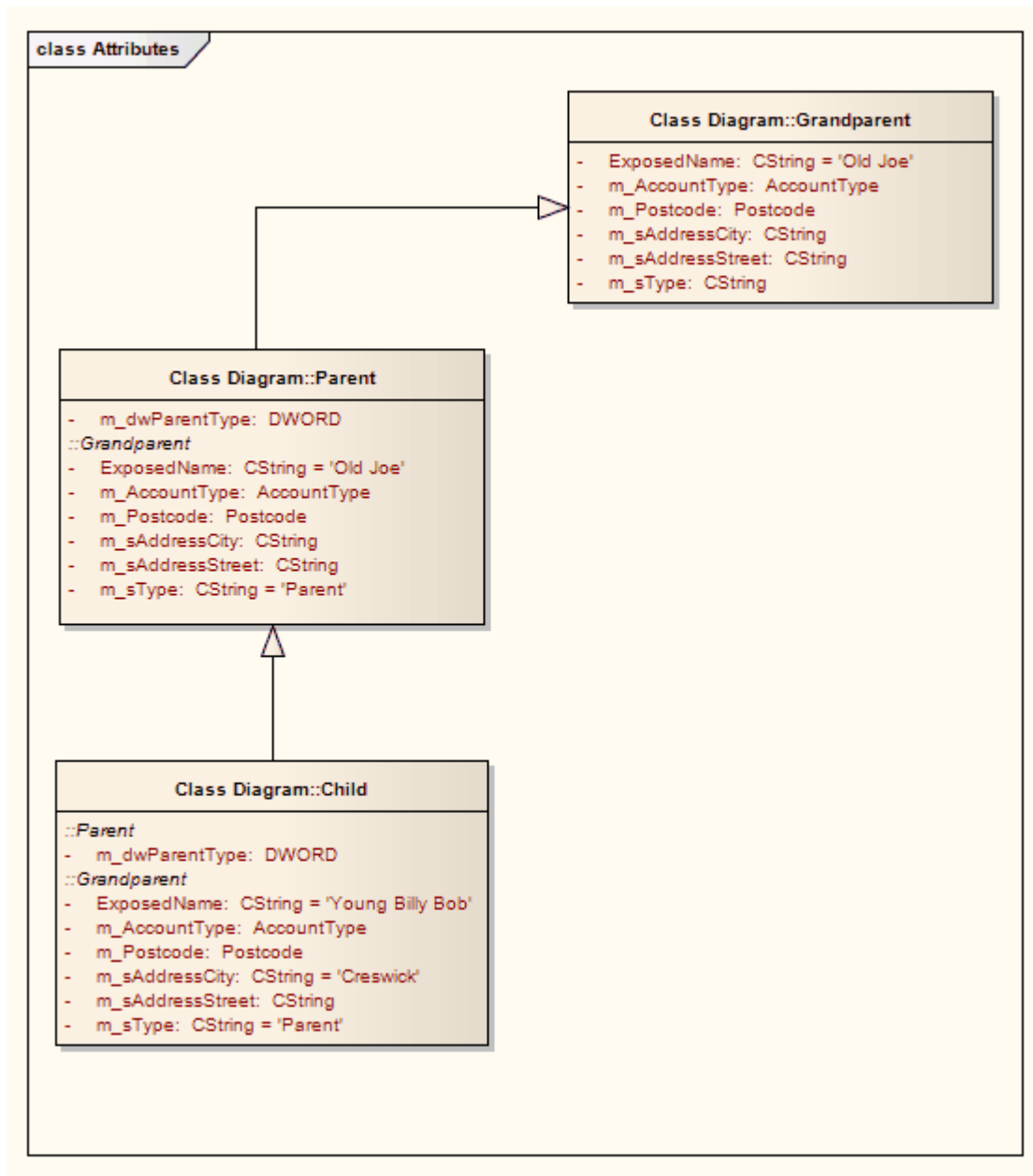
When displaying a Class with attributes in a diagram, you can also show the inherited attributes from all parents in the elements type hierarchy (ancestors).

Access **Right-Click element (on diagram) context menu | Feature Visibility (Ctrl + Shift + Y)**

Topics

Topic	Detail	See also
Override Attribute Initializers	<p>Note that for elements that have attributes you can also override an inherited attribute's initial value, using the element context menu option Advanced Override Attribute Initializers to display the Override Attribute Initializers dialog.</p> <p>In this dialog, select the variable name and enter a new initial value.</p> <p>If required, type a note in the Note field.</p> <p>When you display inherited attributes, Enterprise Architect merges the list of attributes from all ancestors and merges the attribute initializers, so that the final child Class displays the correct attribute set and initial values.</p>	

Example



Learn more

- [Feature Visibility](#)^[745]

5.6.5.8 Create Object From Attribute

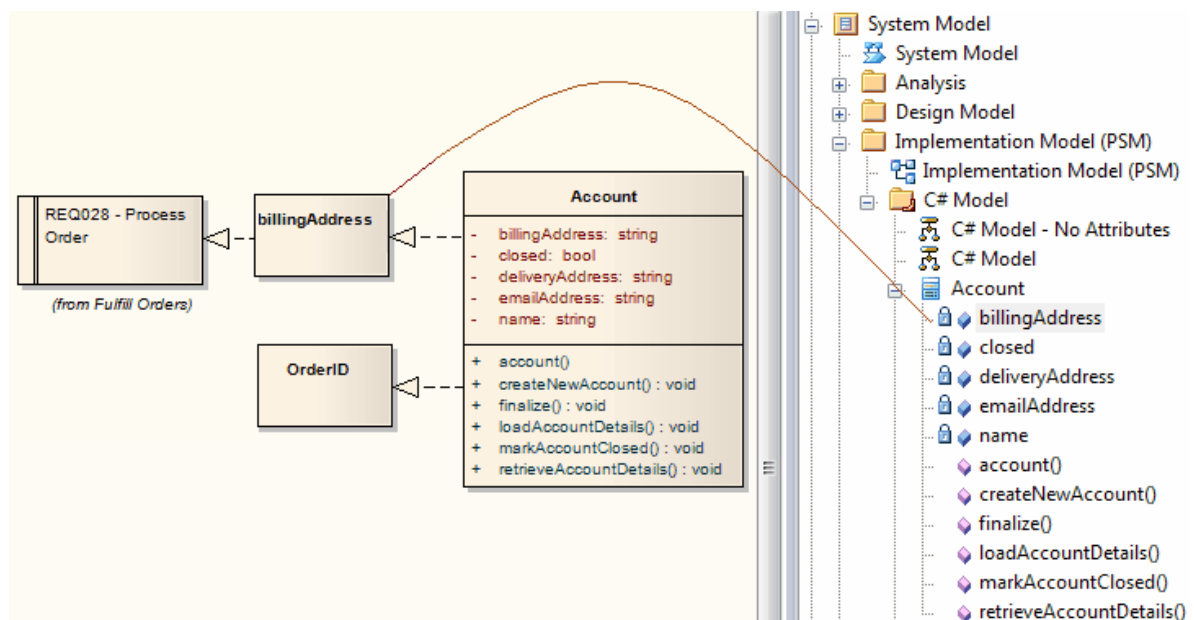
If you drag an attribute from the Project Browser onto an *Activity* diagram, the attribute generates an Object element of the same name.

This is very useful for creating connectors between elements and specific attributes. For example, a Class element of stereotype *table* defines its fields as attributes; Requirement elements that define requirements for those fields can then be linked to the appropriate table fields via the attribute Object elements.

Topics

Topic	Detail	See also
Explanation	<p>In the following diagram, the <i>billingAddress</i> Object was generated by dragging the <i>billingAddress</i> attribute from the Account Class in the Project Browser onto the diagram.</p> <p>The user then created <i>Realize</i> relationships between the <i>Account</i> element and the <i>billingAddress</i> element, and between the <i>billingAddress</i> element and the <i>REQ028 Requirement</i> element.</p>	

Example



Notes

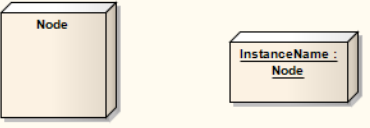
- From Enterprise Architect release 7.5, you can create this relationship more directly by linking an attribute in an element to another element, or linking two attributes in different elements

Learn more

- [Connect to Element Feature](#) ^[944]

5.6.6 Object Classifiers

Topics

Image	Topic	Detail	See also
	<p>Object Classifiers</p>	<p>Many elements model classifications (such as Classes and Actors), whilst other elements model instances of such classifications (such as Objects, Actors again, and Sequence diagram objects).</p> <p>These instance elements represent real things in a run-time scenario; for example, a <i>Person</i> element named <i>Joe Smith</i>; in UML this is written as <i>Joe Smith: Person</i>.</p> <p>You can define a classifier first, and then instances of that classifier; alternatively, as a model develops from a rough sketch to a detailed design, many objects become examples of a defined Class, so in the early analysis phase you might model a <i>Joe Smith</i> and a <i>Jane Smith</i>, and later a <i>Person</i> Class from which <i>Joe</i> and <i>Jane</i> are instantiated.</p> <p>Enterprise Architect enables you to associate an Object with its template element (its classifier), such as a Class; doing this greatly increases the descriptive power of the model in capturing the functionality and responsibility of Objects at run-time and their associated state.</p> <p>For example, if you describe a <i>Person</i> Class with attributes such as <i>Age</i>, <i>Name</i>, <i>Address</i> and <i>Sex</i>, and functions such as <i>GetAge</i> and <i>GetName</i>, then when you associate your Object with the <i>Person</i> Class it is seen to have all the <i>Person</i> Class behavior and state (as well as inherited state and behavior from <i>Person's</i> ancestors).</p> <p>Elements that are classifiers and support instances of themselves at runtime can be dropped from the Project Browser as a link to the classifier itself, or a new instance of the classifier.</p> <p>The example on the left shows a linked Node element and an instance of the Node.</p> <p>Note that the instance is drawn like a simple element with the <code>: <ElementName></code> displayed; if you name your instance it displays <code><InstanceName> : <ElementName></code>.</p>	

Notes

- This is a powerful means of moving your model from the analysis phase into detailed design

5.6.6.1 Using Classifiers

If you right-click on an Object in a diagram, the element context menu displays the **Advanced | Instance Classifier** menu option. Select this option to choose a single element (generally a Class) as the classifier or template for this Object.

The Select <Item> dialog displays. Use this to set the instance classifier.

The Object name is then displayed as *Object: Classifier*; for example a Person object named Joe Smith is displayed as *Joe Smith: Person*.

Topics

Topic	Detail	See also
If an object has a Classifier	<p>It is important to remember that an Object is only an instance of a classifier at runtime, so the appropriate attributes and operations are those of the classifier, not the Object. Therefore, in the context menu for the Object, if you select the Attributes or Operations menu options, the Attributes or Operations dialog displays for the classifier, not the Object.</p> <p>If you set the classifier for an Object in a Sequence diagram, when you add a message the drop-down list of available messages derived from the target Object come from the classifier, not the Object selected. This enables you to associate Sequence diagram objects with Classes and use the defined behavior of the Class to model actual behavior at run time.</p> <p>You can also select a message for a State Flow connector. The same rules apply as for Sequence diagram objects.</p> <p>Note that in the Message dialog you can also select to include messages defined in the target classifier's inheritance hierarchy.</p>	

Learn more

- [Select <Item> Dialog](#)^[880]

5.6.7 Behavior

Enterprise Architect enables you to define an element's behavior through the element's operations and parameters. You can also define the behavior of more specific behavioral elements such as Activities, Interactions, Actions and Interaction Occurrences, through the Behavior and Call tabs of the element Properties dialogs.

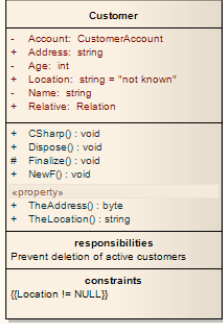
Learn more

- [Operations](#) ^[900]
- [Interactions and Activities](#) ^[910]
- [Behavior Calls](#) ^[910]
- [Behavior Parameters](#) ^[912]
- [Behavior Call Arguments](#) ^[912]

5.6.7.1 Operations

Operations are features of a Class or other element that represent the behavior or services an element supports. For a Customer Class, *UpdateCustomerName* and *GetCustomerAddress* can be operations. Operations have several important characteristics, such as type, scope (visibility), static, abstract and notes.

Topics

Images	Topic	Detail	See also
	<p>Create and Modify Element Operations</p>	<p>In the Diagram view or Project Browser, either:</p> <ul style="list-style-type: none"> • Right-click on the element to be edited, and from the context menu select the Operations menu option, or • Click on the element and press (F10) <p>The <Element name> Operations dialog displays.</p> <p>You can also copy or move existing Operations between elements.</p> <p>These facilities are only available if the target element supports operations.</p>	<p>Operations Dialog - General ^[907]</p> <p>Copy Features Between Elements ^[823]</p> <p>Move Features Between Elements ^[824]</p>
 <pre> classDiagram class Customer { - Account: CustomerAccount + Address: string - Age: int + Location: string = "not known" - Name: string + Relative: Relation + CSharp(): void + Dispose(): void # Finalize(): void + NewF(): void <<property>> + TheAddress(): byte + TheLocation(): string responsibilities Prevent deletion of active customers constraints {{Location != NULL}} </pre>	<p>How Operations Appear in Diagrams</p>	<p>In diagrams, elements with operations (typically Classes) list the operations in the second properties compartment, in colored text, as shown on the left; the default color is dark green (for example, <i>Finalize()</i> : void).</p> <p>Some characteristics display in shorthand form; for example, <i>static</i> displays as \$, <i>abstract</i> as *.</p>	

Images	Topic	Detail	See also
	Operations in the Project Browser	<p>In the Project Browser, Classes with operations have the operations listed beneath the Class name, each preceded by a pink box ().</p> <p>Right-click on an operation and select the Operation Properties context menu option, to open the Operations dialog and edit details for the feature.</p> <p>From the Project Browser, you can drag operations onto new elements to give them the same operations.</p>	

Notes

- If the parent element provides source or target roles for a connector, the connector can be attached to a specific operation

Learn more

- [Behavior Parameters](#) ^[912]
- [Connect to Element Feature](#) ^[944]

5.6.7.1.1 Operations Dialog - General

Access Select element in diagram or Project Browser and:

Element context menu | Operations
(F10) or
Element | Operations

The Operations dialog has seven pages:

- General
- Parameters
- Behavior
- Advanced
- Tagged Values
- Pre-conditions and Post conditions (that is, Constraints)

The General page of the Operations dialog enables you to define new operations and set the most common properties, including name, access type and return.

Field/Button	Usage	See also
Name	Display the selected operation name.	

Field/Button	Usage	See also
Parameters	<p>Display the parameter list.</p> <p>See the <i>Parameters Dialog</i> topic for information regarding what this string can contain.</p>	Parameters Dialog ^[913]
Edit Parameters	Open the Parameters dialog.	
Return Type	<p>(This option is not shown for <i>State</i> or <i>State Machine</i> elements.)</p> <p>Display the data type returned by the operation.</p> <p>The type can be defined by the code language (data type) or by a classifier element; when you click on the drop-down arrow, the set of values in the list provides the appropriate data types.</p> <p>To select or define possible classifiers, either click on the Select Type option in the list, or click on the (...) (Select) button to display the Select <Item> dialog.</p> <p>To add new code language data types that can be displayed in this list, see the <i>Data Types</i> topic.</p>	Instance classifier ^[880] Data Types ^[887]
(...) (Return Type Browse button)	Open the Select <Item> dialog to select the operation return type. (Not shown for <i>State</i> or <i>State Machine</i> elements.)	Instance classifier ^[880]
Action	<p>Define the action of the operation: do, exit or entry.</p> <p>(For <i>State</i> or <i>State Machine</i> elements.)</p>	State ^[1152] State Machine ^[1161]
Scope	<p>Select one of:</p> <ul style="list-style-type: none"> • Public • Protected • Private • Package 	
Stereotype	Specify an optional stereotype for this operation.	
Concurrency	Set the concurrency of the operation.	
Alias	Define an optional alias for the operation.	
Notes	<p>Enter free text notes.</p> <p>You can format this text if necessary, using the Notes toolbar at</p>	Notes toolbar ^[979]

Field/Button	Usage	See also
	the top of the field.	
Virtual/Abstract	If the operation's language is set to C++, map to the C++ <i>Virtual</i> keyword. Otherwise this option is <i>Abstract</i> , pertaining to an abstract function. (This option is not shown for <i>State</i> or <i>State Machine</i> elements.)	
Return Array	Indicate that the return value is an array. (This option is not shown for <i>State</i> or <i>State Machine</i> elements.)	
Synchronized	Specify a code engineering flag that relates to multi threading in Java. (This option is not shown for <i>State</i> or <i>State Machine</i> elements.)	
Static	Indicate that the operation is a static member. (This option is not shown for <i>State</i> or <i>State Machine</i> elements.)	
Const	Indicate that the return type of this method is constant. (This option is not shown for <i>State</i> or <i>State Machine</i> elements.)	
Pure	Indicate that C++ is pure virtual syntax <i>virtual void myFunction() = 0;</i> (This option is not shown for <i>State</i> or <i>State Machine</i> elements.)	
IsQuery	Indicate that this method does not modify the object. (This option is not shown for <i>State</i> or <i>State Machine</i> elements.)	
Up/Down Buttons	Change the order of operations in the list.	
New	Create a new operation.	
Copy	Copy the currently selected operation.	
Save	Save a new operation, or save modified details for existing operation.	

Field/Button	Usage	See also
Delete	Delete the currently selected operation.	
Name/Parameters/ Return Type	List the existing Operations for the selected element.	

Notes

- The General page can vary according to the type of element you are adding an operation to; if defining operations for a data modeling table, see the *Indexes*, *Triggers* and *Check Constraints* topics
- If you make changes and do not save them, the **Cancel** button prompts you to confirm or cancel the changes, whilst the **Close** button closes the dialog immediately and does not save the changes
- If you are creating many operations, go to the Attribute/Operations page of the Options dialog (**Tools | Options | Source Code Engineering | Attribute/Operations**) and select the **After save, re-select edited item** checkbox; now, when you create an operation and click on the **Save** button, the dialog fields clear ready for you to enter the details of the next operation - this helps you when you want to create operations quickly and might not necessarily want to fully define each one as you create it

Learn more

- [Operation Parameters](#) ^[913]
- [Operations Dialog - Behavior](#) ^[904]
- [Operations Dialog - Constraints](#) ^[906]
- [Operation Tagged Values](#) ^[907]
- [Indexes](#) ^[1963]
- [Triggers](#) ^[1965]
- [Check Constraints](#) ^[1966]

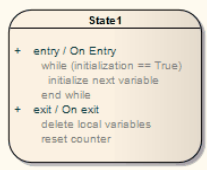
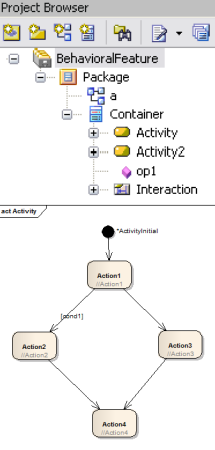
5.6.7.1.1.1 Operations Dialog - Behavior

This topic illustrates how to elaborate a method's function in a diagram. The Behavior page of the Operations dialog enables you to enter free text to describe the functionality of an operation. Use pseudo-code, structured English or just a brief description.

You can also use the Behavior page to formally describe a method or State action and have the text appear under the method/action name in a diagram.

Access **Project Browser Operation context menu | Operation Properties | Behavior**

Topics

Images	Topic	Detail	See also
	State Operations - Behaviour	<p>A State's <i>do</i>, <i>entry</i> and <i>exit</i> operations optionally refer to other behaviors such as Activities, Interactions or Operations.</p> <p>Click on the Element button to display the Select Behavior dialog, and locate and select the required behavior.</p>	Select <Item> Dialog ^[880]
	Associate with Behaviour	<p>A Class operation can be associated with a behavior elsewhere in the model; to do this:</p> <ol style="list-style-type: none"> 1. Display the operation's Properties dialog.. 2. Select the Behavior page, click on the Element button and select the required behavior from the Select Behavior dialog; the behavior displays in the Behavior field. <p>The operation is then set as the specification of the associated behavior.</p> <p>In behavioral code generation, the behavior of the associated behavioral element is generated as the <i>operation's</i> code; in the illustration on the left, <i>Op1</i> is associated with the <i>Activity</i> Activity.</p>	Generate From Behavioral Models ^[1778]
	Example Code	<p>The generated code for <i>Op1</i> is as follows:</p> <pre> package Package; public class Container { public Container(){ } public void finalize() throws Throwable } public void op1(){ /*Activity element(Activity1)'s behavior operation(op1)'s code*/ //Action1; if (cond1) { //Action2; } else { //Action3; } //Action4; } /*Activity element(Activity1)not rendered public void Activity2() { // behavior is a Activity } public void Interaction() { // behavior is a Interaction } </pre>	

Images	Topic	Detail	See also
		}//end Container	

Learn more

- [Initial Code](#)^[906]

Access Project Browser Operation context menu | Operation Properties | Behavior

Topics

Topic	Detail	See also
Usage	<p>On the Behavior page of the Operations dialog, use the Initial Code field to enter code to be inserted into an operation body when the operation is first generated to file.</p> <p>After this point, forward code generation and synchronization do not replace the existing operation code with the Initial Code field.</p> <p>By default, the Initial Code field also is not imported into the model during reverse engineering, but you can select to import the field by selecting the Include method bodies in model when reverse engineering checkbox on the Options dialog.</p> <p>This field is most useful when combined with Design Patterns; elements within a pattern often require the same stub code.</p> <p>Notice that the language-specific patterns available from www.sparxsystems.com/resources/developers/uml_patterns.html include initial code for some of the defined operations; this helps speed up the process of applying patterns from model to implementation.</p> <p>The Initial Code section is also useful for ensuring that the generated code is directly compilable.</p>	<p>Options- Attributes/ Operations^[1899]</p> <p>Design Patterns^[1271]</p> <p>www.sparxsystems.com/resources/developers/uml_patterns.html</p>

5.6.7.1.1.2 Operations Dialog - Constraints

Access Project Browser Operation context menu | Operation Properties | Constraints

Topics

Topic	Detail	See also
Usage	<p>You can define pre- and post- conditions for an Operation; for each type, give the condition a name and a type, and enter notes.</p> <p>Constraints define the contractual behavior of an operation, what must be true before they are called and what is true</p>	

Topic	Detail	See also
	after. In this respect they are related to the state model of a Class and can also relate to the guard conditions that apply to a transition.	

5.6.7.1.2 Operation Tagged Values

Operations can have Tagged Values associated with them. Tagged Values offer a convenient extension mechanism for UML elements, so you can define any tags you like and then assign values to them using this form.

Tagged Values are written to the XML output, and can be input to other third party tools for code generation or other activities.

Access [Project Browser Operation context menu](#) | [Operation Properties](#) | [Tagged Values](#)

How to

To add a Tagged Value for an operation

Step	Action	See also
1	Click on the Tagged Values page of the operation Properties dialog.	
2	Click on the New Tags button. The Tagged Value dialog displays.	
3	In the Tag field, type the tag name (or select a defined tag from the drop-down list). In the Value field type the initial tag value.	
4	Click on the OK button to confirm the operation.	

Notes

- Tagged Values are supported for attributes, operations, objects and connectors
- You can define custom tags using a predefined Tagged Value Type

Learn more

- [Create a Custom Tagged Value](#)^[393]

5.6.7.1.3 Override Parent Operations

In Enterprise Architect, you can automatically override methods from parent Classes and from realized Interfaces.

Access [Element](#) | [Advanced](#) | [Overrides & Implementations](#)

Topics

Topic	Detail	See also
Usage	<p>In the Override Operations/Interfaces dialog, check the operations/interfaces to automatically override and click on the OK button.</p> <p>Enterprise Architect generates the equivalent function definitions in your child Class.</p> <p>You can configure Enterprise Architect to display this dialog each time you add a Generalization or Realization connector between Classes, and review their possible operations/interfaces to override/implement; to do this, open the Links page of the Options dialog (select the Tools Options Links menu option).</p>	

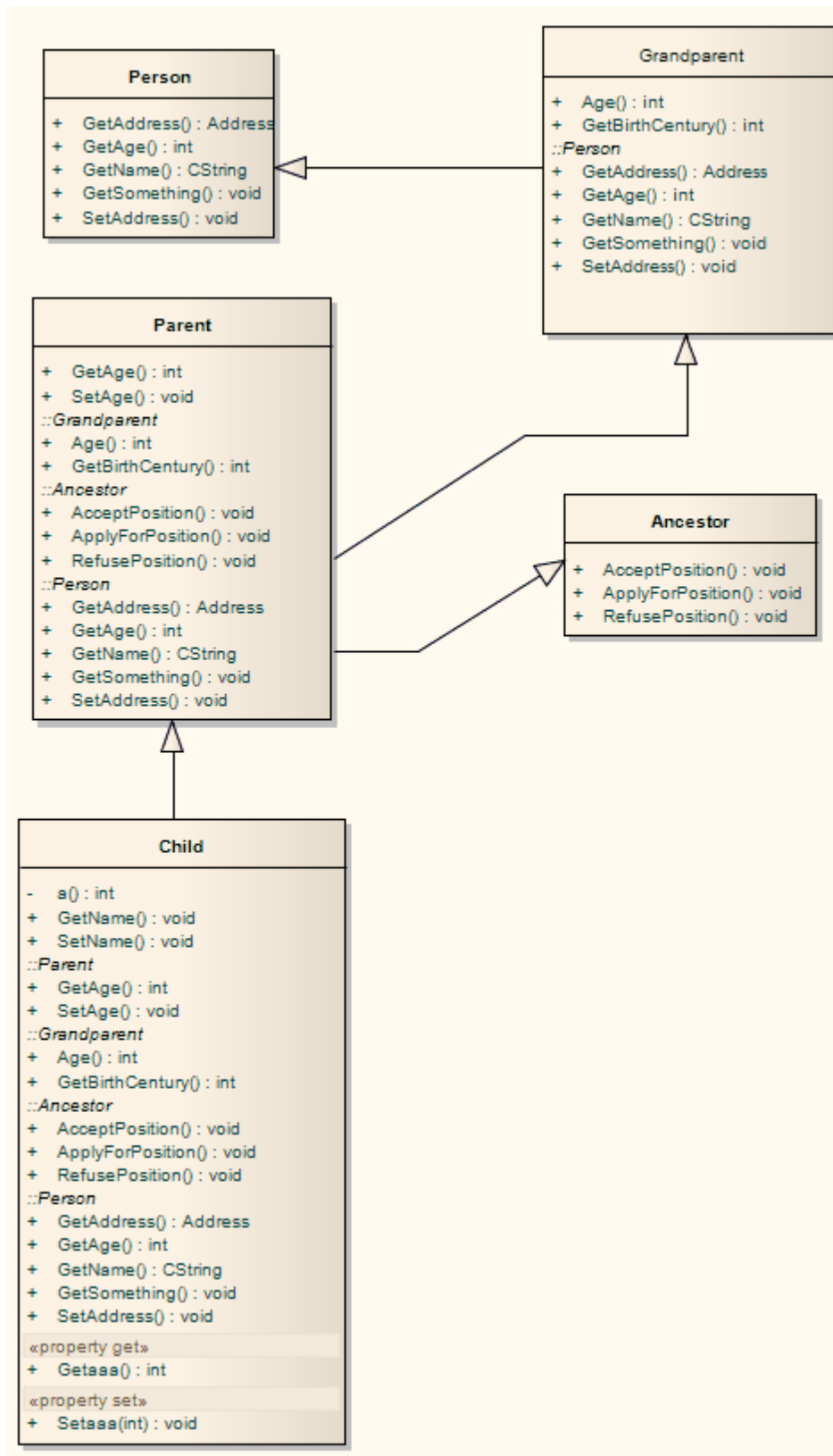
5.6.7.1.4 Display Inherited Operations

You can configure an element in a diagram to display the complete operation set obtained from all ancestors in the element's type hierarchy, as well as those directly owned.

Access [Element](#) | [Feature Visibility](#) | **Shortcut: (Ctrl+Shift+Y)**

Example

The following diagram illustrates this behavior when enabled for each element in a simple hierarchy.



5.6.7.2 Interactions and Activities

The behavioral aspects of Interactions and Activities are modeled using the Behavior page of the element Properties dialog, which enables you to assign parameters and return types to the elements.

Topics

Topic	Detail	See also
Usage	<p>Use the Edit Parameters button to edit an element's parameters. When you create a new parameter using the dialog, it internally creates an Activity Parameter Node for an Activity or an Interaction Parameter for an Interaction. In the Return field, click on the drop-down arrow and select the return type of the element.</p> <p>The specification field is populated automatically when an operation is associated with the activity as a behavior.</p>	<p>Parameters Dialog^[913]</p> <p>Activity Parameter Nodes^[1103]</p> <p>Operations Dialog - Behaviour^[904]</p>

5.6.7.3 Behavior Calls

A behavior call is the invocation of a behavior. You can represent an invocation with a Call Operation Action (Operation), Call Behavior Action (Activity) or Interaction Occurrence (Interaction) element. You model the properties of the behavior call using the Call tab of the element Properties dialog.

Use to

- Edit Arguments
- Re-associate the call with a different behavior
- Synchronize the arguments with the parameters in the associated behavior

Topics

Topic	Detail	See also
Usage	<p>Click on the Edit button to create and delete arguments, and relate them to a corresponding parameter in the associated behavior.</p> <p>Click on the (...) (Select Behavior) button to re-associate the invocation with a different behavior or to remove any association with the current behavior.</p> <p>The Interaction Occurrence panel is displayed only for Interaction Occurrence elements. It enables you to enter the return value and attribute of the behavior call.</p>	

Learn more

- [Action Notation](#)^[1095]

- [Interaction Occurrence](#)^[1139]
- [Arguments](#)^[912]
- [Associate with Different Behavior](#)^[911]
- [Synchronize Arguments](#)^[911]
- [Code Generation - Activity Diagrams](#)^[1787]

5.6.7.3.1 Associate with Different Behaviors

On the Call tab of the Behavior Call Properties dialog, when you click on the (...) (Select Behavior) button the Select <Item> dialog displays, listing all available behaviors in the model.

Topics

Topic	Detail	See also
Usage	<p>Select <none> to remove any existing association between an invocation and a behavior, or select another classifier to re-associate the invocation with a different behavior.</p> <p>The Synchronize with Parameters button is enabled only if a valid behavior is identified in the Behavior field.</p>	

Learn more

- [Select <Item> Dialog](#)^[880]

5.6.7.3.2 Synchronize Arguments

Topics

Topic	Detail	See also
Usage	<p>On the Call tab of the element Properties dialog, click on the Synchronize with Parameters button to synchronize the number of arguments in the invocation element with the number of parameters in the associated behavior.</p> <p>This automatically creates or deletes arguments based on the number of parameters in the behavior.</p> <p>If any arguments are to be deleted, Enterprise Architect prompts you to confirm the operation. Click on the Yes button to confirm.</p>	

Notes

- The **Synchronize with Parameters** button is enabled only if the invocation is associated with a valid behavior, as identified in the **Behavior** field

5.6.7.4 Behavior Call Arguments

You define the arguments of a Behavior Call using the Arguments dialog, which you display by clicking on the **Edit** button on the Call tab of the element Properties dialog.

How to

To define the arguments of a Behavior Call

Step	Action	See also
1	In the Name field, type the name of an argument to map to the behavior.	
2	In the Parameters field, click on the drop-down arrow and select a behaviors parameter from the list of parameters associated with the behavior.	
3	In the Value field, set any required value.	
4	If a diagram is displayed, and if required, select the Show in current diagram checkbox to add an Action Pin on the diagram.	Action Pin <small>[1097]</small>
5	Click on the Save button.	
6	If appropriate, click on the New button and repeat steps 1 to 5 for another argument: parameter mapping.	

Notes

- If you attempt to map a newly created argument to a parameter that is already associated with a different argument, Enterprise Architect identifies the mapping and prompts you to confirm that you intend to change the association

Learn more

- [Behavior Calls](#) [910]

5.6.7.5 Behavior Parameters

This topic area describes the facilities for defining, editing and extending the parameters of behavioral operations, Activities and Interactions.

Learn more

- [Parameters Dialog](#) [913]
- [Parameter Tagged Values](#) [914]

- [Operation Parameters By Reference](#)^[915]

5.6.7.5.1 Parameters Dialog

The Parameters dialog enables you to define the parameters of an operation, Activity or Interaction. The parameter list is reproduced in code in the order the parameters appear in the parameters list, so use the **Up** and **Down** buttons to move parameters into their required positions. Additionally, you can select the **Add new to end** checkbox to force new parameters to appear at the end of the list instead of the top.

Reference

Field	Usage	See also
Name	Indicates the parameter name.	
Type	Select the data type of the parameter. Alternatively, click on the (...) button and select the element classifier to define the type.	
Default	Indicates an optional default value for the parameter.	
Stereotype	Indicates a stereotype name, specification of a Stereotype name can also be chosen via the drop-down list.	
Kind	Indicate the way a parameter is passed to a function: <ul style="list-style-type: none"> • In = By Value • InOut = By Reference • Out is passed by Reference, but only the return value is significant 	
Fixed	Set the parameter to <i>const</i> , even if passed by reference.	
Alias	Indicate an optional alias for the parameter.	
Add new to end	Place new parameters at the end of the list instead of the start.	
Multiplicity	Display the Multiplicity dialog, to specify the multiplicity of the parameters.	
Notes	Indicate any additional notes on the parameter.	

Below is information regarding the Multiplicity dialog.

Reference

Field	Usage	See also
Lower bound	Define a lower limit to the number of elements allowed in the collection.	
Upper bound	Define an upper limit to the number of elements allowed in the collection.	
Allow Duplicates	Indicate that duplicates are allowed. This maps to the UML property <i>isUnique</i> , value FALSE .	
Multiplicity is Ordered	Indicate that the collection is ordered.	

Notes

- Set the amount of parameter detail to display in a specific diagram using the **Show Parameter Detail** drop-down list on the Diagram Properties dialog; the setting applies only to the current diagram, and the default is to show the type only

Learn more

- [Parameter Tagged Values](#)^[914]
- [Operation Parameters by Reference](#)^[915]
- [Visible Class Members](#)^[732]

5.6.7.5.1.1 Parameter Tagged Values

Behavioral parameters can have Tagged Values associated with them. Tagged Values offer a convenient extension mechanism for UML elements; you can define any tags you like and then assign values to them using this form.

Tagged Values are written to the XML output, and can be input to other third party tools for code generation or other activities.

How to

To add a Tagged Value for a parameter

Step	Action	See also
1	Double-click on the operation, Activity or Interaction containing the parameter in a diagram or in the Project Browser.	

Step	Action	See also
	The Properties dialog displays.	
2	Click on the Tagged Values tab, which shows the Tagged Values for the selected object and its parameters.	
3	Click on the required parameter in the Parameters compartment of the Tagged Values tab, and click on the New Tags button. The Tagged Value dialog displays.	
4	In the Tag field, type the tag name (or select a defined tag from the drop-down list). In the Value field type the initial tag value.	
5	Click on the OK button to confirm the Tagged Value.	

Notes

- Tagged Values are supported for attributes, operations, objects and connectors
- Custom tags can be created using a predefined Tagged Value Type

Learn more

- [Create a Custom Tagged Value](#) ^[1393]

5.6.7.5.2 Operation Parameters by Reference

Access **Tools | Options | Objects : Feature reference indicator**

Topics

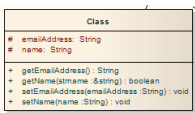
Image	Topic	Detail	See also
 <pre> Class # emailAddress: String # name: String + getEmailAddress(): String + getName(stname: String): boolean + setEmailAddress(emailAddress: String): void + setName(name: String): void </pre>	Usage	<p>You can select to highlight parameters declared as 'Kind: <i>inout</i>' or 'Kind: <i>out</i>' with an additional user-defined prefix or suffix.</p> <p>If you select the Highlight References checkbox, you can also indicate whether a prefix or suffix should be used, and the actual reference character to use. In the example above, the & character has been set as a prefix.</p> <p>When you declare a parameter of type <i>inout</i>, it is assumed you are passing the parameter by reference rather than by value. If you have elected to</p>	Objects ^[556]

Image	Topic	Detail	See also
		<p>highlight references, then this is displayed in the Diagram View.</p> <p>The example to the left shows that, in the <i>getName</i> operation, the parameter <i>strName</i> is a string reference, and is highlighted using the chosen character and position.</p>	

Notes

- This facility currently applies to operations only

5.6.8 In-place Editing Options

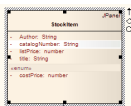
This topic explores the tasks that can be performed using in-place editing of elements on a diagram in Enterprise Architect. The tasks include:

- [View Properties](#)^[917]
- [Edit Element Item Name](#)^[918]
- [Edit Feature Stereotype](#)^[919]
- [Edit Feature Scope](#)^[920]
- [Edit Attribute Keyword](#)^[921]
- [Edit Operation Parameter Keyword](#)^[922]
- [Insert Operation Parameter](#)^[924]
- [Edit Parameter Kind](#)^[923]
- [Insert New Feature](#)^[924] (Attribute or Operation)
- [Insert Maintenance Feature](#)^[925]
- [Insert Testing Features](#)^[926]
- [Delete Selected from Model](#)^[917]

5.6.8.1 In-place Editing Tasks

How to

To use the in-place editing menu options

Images	Step	Action	See also
	1	Open the diagram containing the element.	
	2	<p>Click on the element, and on the item to manipulate within the element.</p> <p>The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.</p>	

Images	Step	Action	See also
	3	<p>Edit and manipulate the items in the element, either by pressing the appropriate keyboard keys or by right-clicking on the highlighted item and choosing a task from the Element Items context menu.</p> <p>The available commands are listed in the table below.</p>	

Reference

Menu option...	Usage	Shortcut	See also
Edit Selected	Change the name, scope or stereotype of the element or element item.	F2	
View Properties	Display the dialog containing details of the element.	Enter	
Insert New After Selected	Insert a new item in the element.	Insert	
Find in Project Browser	Locate the item in the Project Browser.		
Add Attribute	Add an attribute to the element.	Ctrl + Shift + F9	
Add Operation	Add an operation to the element.	Ctrl + Shift + F10	
Add Other	Insert a feature on the specific element item, such as Maintenance features and Testing features.	Ctrl + F11	
Delete Selected from Model	Delete the selected item from the model.	Delete	
View Source code	Display the source code for the element, in the default external editor.	F12	Editing Source Code ^[1799]
Set Breakpoint	Set a breakpoint on a highlighted operation (including a breakpoint, start recording marker, end recording marker or stack auto-capture marker.)		Breakpoint and Marker Management ^[1869] Marker Types ^[2155]
	Navigate Diagram Selection, to navigate the	Ctrl + Shift +	

Menu option...	Usage	Shortcut	See also
	diagram between elements without having to use the mouse.	arrow key	
	Toggle element highlight option on and off.	Shift + Enter	

Other options that are available while editing element attributes or operations in a diagram include:

Menu options	Usage	Shortcut	See also
	Accept current changes.	Enter	
	Accept current changes and open a new slot to add a new item.	Ctrl + Enter	
	Abort the edit, without save.	Esc	
	Move the cursor between segments of a feature definition.	Tab or Shift + Tab	
	Display the context menu for in-place editing.	Shift + F10	
	Invoke the Select <Item> dialog.	Ctrl + Shift + Space	Select <Item> Dialog ^[880]
	Display an autocompletion list for a feature classifier.	Ctrl + Space	



Notes

- Most of the in-place editing menu commands have keyboard alternatives; for many of them, if the selected item happens to be off-screen when you press the appropriate keys, the diagram automatically scrolls to show the whole element, so that you can see what you are changing

5.6.8.2 Edit Element Item Name

How to


To change the name of the element, or the name of an operation or attribute, directly from the diagram, follow the steps below


Images	Step	Action	See also
	1	Open the diagram containing the element.	
	2	Click on the element and on the name to change within the element. The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Right-click on the item. The context menu displays.	
	4	Select the Edit Selected menu option, or press (F2) to enable you to edit the item directly from the diagram. The name of the attribute or operation is highlighted.	
	5	Delete or type over the name.	
	6	Press (Enter) to accept the change, or (Esc) to cancel the change.	

5.6.8.3 Edit Feature Stereotype

How to

To change the stereotype of an operation or attribute directly from the diagram

Images	Step	Action	See also
	1	Open the diagram containing the element.	
	2	Click on the element, and on the item to edit within the element. The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Right-click on the item. The context menu displays.	
	4	Select the Edit Selected menu option (or press (F2)) to enable you to edit the attribute or operation directly from the diagram. The name of the item is highlighted.	

Images	Step	Action	See also
	5	Move the cursor to the position before the name or within the existing attribute or operation stereotype (denoted by «stereotype name»).	
	6	Delete or type over the previous name to change the stereotype name of the attribute or operation.	
	7	Press (Enter) to accept the change or (Esc) to cancel the change.	



Notes

- You can assign multiple stereotypes by including a comma-separated list inside the stereotype markers

5.6.8.4 Edit Feature Scope

How to

To rapidly change the scope of an attribute or operation directly from the diagram

Images	Step	Action	See also
	1	Open the diagram containing the element.	
	2	Click on the element and on the item to edit within the element. The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Right-click on the item. The context menu displays.	
	4	Select the Edit Selected menu option (or press (F2)) to enable you to edit the attribute or operation directly from the diagram. The name of the item is highlighted.	
	5	Move the cursor to the scope of the item and delete the previous entry.	
	6	Reassign the entry by typing in one of the following symbols: <ul style="list-style-type: none"> + indicates that the scope is Public 	


Images	Step	Action	See also
		<ul style="list-style-type: none"> • - indicates that the scope is Private • ~ indicates that the scope is Package • # indicates that the scope is Protected 	
	7	<p>Press (Enter) to save the change, or (Esc) to cancel the change.</p> <p>The diagram is updated to reflect the changes (also see the <i>catalogNumber</i> attribute in the screen illustrations to the left).</p>	

5.6.8.5 Edit Attribute Keyword

You can add features such as attribute keywords and classifiers directly to an element, using the **Element Keywords and Classifiers** menu. This enables you to rapidly assign details element item by element item, directly from a diagram.

How to

To add features directly to an element

Images	Step	Action	See also
	1	In Enterprise Architect, open the diagram containing the element.	
	2	<p>Click on the element, and on the attribute to edit within the element.</p> <p>The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.</p>	
	3	<p>Right-click on the item.</p> <p>The context menu displays.</p>	
	4	<p>Select the Edit Selected menu option (or press (F2)) to enable you to edit the attribute directly from the diagram.</p> <p>The name of the attribute is highlighted.</p>	
	5	Right-click on the attribute name to display the context menu.	
	6	<p>From the context menu, you can:</p> <ul style="list-style-type: none"> • Change the attribute classifier to static or fixed - select the static or fixed menu options as appropriate; the diagram is updated to reflect the changes 	Properties Dialog ^[843]


Images	Step	Action	See also
		<ul style="list-style-type: none"> Display the Class properties - click on the Goto Definition menu option; Enterprise Architect locates the Class in the Project Browser and opens its Properties dialog <p>If the data type is a raw data type, Enterprise Architect displays the message: <i>The data type is a raw data type</i></p>	

5.6.8.6 Edit Operation Parameter Keyword

You can directly edit operation classifiers by element, using the in-place editing menu. This enables you to rapidly assign parameter keywords.

How to

To directly edit operation classifiers by element

Images	Step	Action	See also
	1	Open the diagram containing the element.	
	2	Click on the element, and on the operation to edit within the element. The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Right-click on the item. The context menu displays.	
	4	Select the Edit Selected menu option (or press (F2)) to enable you to edit the operation directly from the diagram. The name of the operation is highlighted.	
	5	Right-click on the data type of a parameter to display the context menu.	
	6	From the context menu you can: <ul style="list-style-type: none"> Change the operation classifier by clicking on the appropriate menu option - static, isquery, abstract or fixed; the diagram is updated to reflect the changes Display the Class properties - click on the Goto Definition menu option 	Properties Dialog ^[843]


Images	Step	Action	See also
		<p>If the data type is Class, Enterprise Architect locates the Class in the Project Browser and opens its Properties dialog</p> <p>If the data type is a raw data type, Enterprise Architect displays the message <i>This data type is a raw data type</i></p> <p>If the data type is not defined in the model, the message is: <i>The data type is not defined in the model</i></p>	

5.6.8.7 Edit Parameter Kind

You can edit operation parameter kinds such as *(in)*, *(inout)* and *(out)* directly from a diagram element by element, using the **Element Keywords and Classifiers** menu. This enables you to rapidly assign the parameter directly from a diagram.

How to

To edit operation parameter kinds directly from a diagram


Images	Step	Action	See also
	1	In Enterprise Architect, open the diagram containing the element.	
	2	Click on the element, and on the operation to edit within the element. The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Right-click on the item. The context menu displays.	
	4	Select the Edit Selected menu option (or press (F2)) to enable you to edit the item directly from the diagram. The name of the item is highlighted.	
	5	Right-click on the item name to display the context menu.	
	6	Select the appropriate menu option for the parameter kind value: (in), (inout) and (out). The diagram is updated to reflect the change.	

Images	Step	Action	See also

5.6.8.8 Insert New Feature

How to

To add attributes and operations to a Class diagram element


Images	Step	Action	See also
	1	Open the diagram containing the element to which you are adding an attribute or operation.	
	2	Click on the element and, within the element, on the item after which to insert the operation or attribute. The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Either: <ul style="list-style-type: none"> • Press (Insert) or • Right-click on the selected element item to display the context menu and select the Insert New After Selected menu option Enterprise Architect inserts a new data line in the diagram, underneath the selected item.	
	4	Type in the relevant information for the attribute or operation. Use the (Tab) and (Tab+Shift) keys to move the cursor through the segments of the attribute or operation, and select the classifier from either the: <ul style="list-style-type: none"> • Autocompletion list (press the (Ctrl+Space) keys) or • Select <Item> dialog (press the (Ctrl+Shift+Space) keys) 	Autocompletion List ^[1812] Select <Item> Dialog ^[880]
	5	Press (Enter) to accept the change or (Esc) to cancel the change. The diagram is updated to reflect the changes.	

5.6.8.9 Insert Operation Parameter

You can add operation parameters to an operation through the in-place editing options, using hotkey commands or menu shortcuts.

How to

To add parameters to operations in a Class diagram element

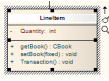
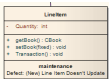
Images	Step	Action	See also
	1	Open the diagram containing the element.	
	2	Click on the element, and on the operation to update within the element. The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Press (F2), or right-click on the selected item to display the context menu and select the Edit Selected option.	
	4	Move the cursor inside the parameter brackets and type the parameter name, followed by a colon (for example, bks: for a vector containing books).	
	5	Give the parameter a type. Place the cursor after the colon at the end of the name and add the classifier; either: <ul style="list-style-type: none"> • Press (Ctrl+Space) to invoke the classifier autocompletion list • Press (Ctrl+Shift+Space) to select a classifier from the Select <Item> dialog or • Right-click the mouse to display the inline editing options context menu and select the Insert Classifier option <p>The Select <Item> dialog displays.</p>	Autocompletion List ^[1812] Select <Item> Dialog ^[880]
	6	Press (Enter) to accept the change or (Esc) to cancel the change. The diagram is updated to reflect the changes.	

5.6.8.10 Insert Maintenance Feature

How to

To rapidly assign maintenance details such as Defects, Changes, Issues and Tasks directly to an element from a diagram

Images	Step	Action	See also
	1	Open the diagram containing the element.	

Images	Step	Action	See also
	2	Click on the element name. The name is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Either: <ul style="list-style-type: none"> • Press (Ctrl+F11) or • Right-click on the highlighted name to display the context menu, and select the Add Other option The Insert Feature dialog displays.	
	4	Click on the appropriate radio button option to associate the required maintenance feature with the element item.	
	5	Click on the OK button. The <Maintenance Feature> details for <element> dialog displays.	
	6	Complete the fields to define the maintenance activity, and then click on the Apply button. To create a subsequent maintenance activity of this type, click on the New button.	
	7	When you have defined all of the maintenance activities of this type, click on the OK button. The maintenance details are added to the element.	

Notes

- To ensure that the maintenance items are visible in the diagram element, as shown in step 7, select the **Maintenance** checkbox on the Elements tab of the Diagram Properties dialog



Learn more

- [Show Maintenance Script in Diagram](#) ^[2242]

5.6.8.11 Insert Testing Features

How to

To rapidly add testing features such as Unit, Integration, System, Acceptance and Scenario tests to an element directly from a diagram

Images	Step	Action	See also
	1	Open the diagram containing the element.	
	2	Click on the element. The element name is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Either: <ul style="list-style-type: none"> • Press (Ctrl+F11) or • Right-click on the highlighted name to display the context menu and select the Add Other option The Insert Feature dialog displays.	
	4	Click on the appropriate radio button option to associate the required testing feature with the element.	
	5	Click on the OK button. The Testing window opens, showing the appropriate panel for the type of test selected.	
	6	Complete the fields to define the test activity, and then click on the Save icon in the window toolbar. The test is added to the element.	The Testing Workspace ^[2213]
	7	To create a subsequent test activity of this type, click on the New icon. To add items for other types of test, click on the appropriate tab.	

Notes

- To ensure that the test items are visible in the diagram element, as shown in step 7, select the **Testing** checkbox on the Elements tab of the Diagram Properties dialog

Learn more

- [Show Test Script Compartments](#) ^[2229]

5.6.9 Linked Documents

In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, you can link an RTF document to any UML element in the model.

All editions of Enterprise Architect provide an additional UML Artifact - Document Artifact - that can contain an RTF document internally.

You create linked documents from Linked Document Templates, which you define with the Document Template Editor.

Topics

Topic	Detail	See also
Usage	<p>Documents created via the Document Artifact element are rendered into RTF Documentation by selecting the Linked Document checkbox in the RTF Style Template Editor.</p> <p>The Linked Document checkbox is within the Element hierarchy, towards the end. Remember that checkboxes can be moved up and down the hierarchy to position information in the generated document as you require. In some templates, the Linked Document checkbox is only available as a child of the External Requirements checkbox.</p> <p>The linked document is rendered into the RTF documentation at:</p> <p><i>linked document ></i></p> <p><i><linked document</i></p>	<p>Reporting Linked Documents ^[2309]</p>

Notes

- When you have saved the document, an **A** symbol displays in the bottom right corner of the element

Learn more

- [Document Artifact](#) ^[1191]
- [Create Linked Document Templates](#) ^[934]
- [Edit Linked Document Templates](#) ^[934]
- [Select Components For Documentation](#) ^[2304]
- [Create Document Artifact](#) ^[929]
- [Link Document to UML Element](#) ^[929]
- [Edit Linked Documents](#) ^[930]
- [Hyperlink From Linked Document](#) ^[931]
- [Create Element From Document](#) ^[932]
- [Replace or Delete Documents](#) ^[933]
- [Generate RTF Documentation Dialog](#) ^[2256]

5.6.9.1 Create Document Artifact

Topics

Topic	Detail	See also
Usage	<p>You create a Document Artifact element in a Component or Deployment diagram.</p> <p>Drag and drop the Document Artifact element from the Toolbox into your diagram.</p> <p>Double-click on the Document Artifact element. The Linked Document Editor opens, with the New Linked Document dialog.</p> <p>In the Copy template field, click on the drop-down arrow and select a previously-created Linked Document Template. Click on the OK button.</p>	<p>Component Diagram ^[1023]</p> <p>Deployment Diagram ^[1020]</p> <p>Linked Document ^[930]</p> <p>Create Linked Document Templates ^[934]</p> <p>Edit Linked Document Templates ^[934]</p>

5.6.9.2 Link Document to UML Element

How to

Step	Action	See also
1	<p>Click on an element in the Project Browser, Diagram List, Package Browser, Model Search or diagram, and either:</p> <ul style="list-style-type: none"> • Select the Element Linked Document menu option • In the Notes window, click on the Linked Document icon in the toolbar • Press (Ctrl+Alt+D) or • Right-click and select the (Create) Linked Document option from the context menu <p>Alternatively, open the element Properties dialog and click on the Linked Document icon in the toolbar in the Notes field.</p>	
2	The New Linked Document dialog displays.	
3	Select the previously-created template from which to create the document.	
4	<p>Click on the OK button.</p> <p>The Linked Document editor displays.</p>	Linked Document Editor ^[930]
5	Enter the text of the document.	

Step	Action	See also

Notes

- This operation is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- When you have saved the document, an **A** symbol displays in the bottom right corner of the element

Learn more

- [Create Linked Document Templates](#)^[934]
- [Edit Linked Document Templates](#)^[934]

5.6.9.3 Edit Linked Documents

Topics

Topic	Detail	See also
Usage	<p>Enterprise Architect provides a Windows-like word processor to help you edit Linked Documents. This is the same as the RTF Style Template Editor, and it provides the same convenient features.</p> <p>You can format and edit the document in a number of ways, and add links from and references to the content of the document.</p> <p>For example, you can highlight a word or term in the linked document and select Create Glossary Definition to create a glossary definition for that term. Anyone reading the document can, if they check for the term in the Project Glossary, read the definition.</p>	
Toolbars	<p>The Linked Document Editor has two toolbars of standard icons that also enable you to create, save, print and edit your linked documents. These toolbars display immediately above the editing window.</p> <p>You can toggle the display of these toolbars by clicking on the Show/Hide Linked Document toolbar icon (third from the right) in the View header bar.</p>	

Learn more

- [Create a Glossary Definition](#)^[468]
- [Glossary Dialog](#)^[469]
- [Scroll Through Text](#)^[2276]

- [File and Print Options](#) ^[2277]
- [Cut and Paste Options](#) ^[2279]
- [Image and Object Imports](#) ^[2282]
- [Character Formatting](#) ^[2284]
- [Paragraph Formatting](#) ^[2285]
- [Tab Support](#) ^[2288]
- [Page Breaks and Repagination](#) ^[2289]
- [Insert Headers and Footers](#) ^[2290]
- [Insert Bookmarks](#) ^[2291]
- [Table Commands](#) ^[2292]
- [Sections and Columns](#) ^[2296]
- [Stylesheets and Table of Contents](#) ^[2297]
- [Text/Picture Frame and Drawing Objects](#) ^[2302]
- [Search/Replace Commands](#) ^[2303]
- [Hyperlink From Linked Document](#) ^[931]
- [Create Elements From Linked Documents](#) ^[932]

5.6.9.4 Hyperlink From Linked Document

Topics

Topic	Detail	See also
How To	<p>To create a hyperlink from a linked document, click on the object in the Project Browser and drag it to the point at which to create the hyperlink. The linked document editor automatically creates the hyperlink, using the object name as the hyperlink text. You can edit this text if required.</p> <p>Similarly, you can create a hyperlink to an element in the model by highlighting the link text in the linked document, right-clicking on it and selecting the Create Link to Existing Element context menu option. This displays the Select Classifier dialog, from which you select the element to link to.</p> <p>In either case, when you next open the document, you can double-click on the hyperlink to locate and highlight the object in the Project Browser. You can then perform all normal operations on the object, including opening any linked document on the highlighted element.</p> <p>You can also create a hyperlink to a wide range of additional objects, such as web pages, Help files, Model Searches and Team Review Forums, by highlighting the appropriate text and then selecting the Create New Hyperlink context menu option. This displays the Hyperlink Details dialog. When specifying the object location, you can use full paths or local (path substitution) paths.</p> <p>For an alternative method of creating a hyperlink to an external document, Help file or web page, see the <i>Hyperlinks and Bookmarks</i> topic.</p>	<p>Select <Item> Dialog ^[880]</p> <p>Model Searches ^[621]</p> <p>Team Review ^[279]</p> <p>Hyperlinks ^[1668]</p> <p>Hyperlinks and Bookmarks ^[2291]</p>

5.6.9.5 Create Element From Document

Using the Linked Document Editor, you can create document-specific elements and diagrams in the Project Browser, with hyperlinks from the document to the created item. When you click on the hyperlink, the element or diagram is highlighted in the Project Browser. The element or diagram is created in the same package as the element for which the linked document was created.

Topics

Topic	Detail	See also
Usage	<p>You can create and link to any type of element or diagram, but the facility has specific options for the following element types:</p> <ul style="list-style-type: none"> • Class • Requirement • Issue <p>You can create the same arrangement with existing elements, diagrams and packages by dragging them from the Project Browser into the text of the document, creating a hyperlink with the item name as the text.</p>	<p>Class ^[118]</p> <p>Requirement ^[147]</p> <p>Issue ^[224]</p> <p>Hyperlink From Linked Document ^[93]</p>

How to

Step	Action	See also
1	Open the linked document, either from a Document Artifact element or through the context menu for an existing element (Corporate, Business and Software Engineering, System Engineering and Ultimate editions).	<p>Document Artifacts ^[92]</p> <p>Linking Document to UML Element ^[92]</p>
2	Enter some text, including appropriate text to act as the link (such as the element or diagram name).	
3	Highlight the appropriate text and right-click on it. The editor context menu displays.	
4	<p>Select the Create New menu option, and the required submenu option.</p> <p>If you select the:</p> <ul style="list-style-type: none"> • Class, Requirement or Issue option, the corresponding element is immediately created in the Project Browser • Other option, the New Element dialog displays; specify the element type and - if appropriate - stereotype, and click on the Create button • Diagram option, the New Diagram dialog displays; specify the diagram 	<p>New Elements ^[79]</p> <p>New Diagrams ^[72]</p>

Step	Action	See also
	type and click on the OK button	
5	The highlighted text is now a hyperlink. Click on the link to highlight the new element or diagram in the Project Browser. You can now edit or expand the element or diagram as required.	

5.6.9.6 Replace or Delete Documents

If a linked document is out of date, you can either edit the text or replace the entire contents from another file.

How to

To replace the contents

Step	Action	See also
1	Click in the body of the document and press (Ctrl+A) to select all the document text.	
2	Press (Delete).	
3	Right-click and select the File Import context menu option. The Windows Open dialog displays, in which you can browse for the file to import into the document.	
4	Click on the Save icon in the Linked Document screen toolbar.	

To delete the linked document

Step	Action	See also
1	Click on an element in the Project Browser or diagram, and either: <ul style="list-style-type: none"> • Select the Element Delete Linked Document menu option or • Right-click and select the Delete Linked Document context menu option 	
2	Enterprise Architect prompts you to confirm the deletion; click on the Yes button. If required, you can now create another linked document for the element.	

Learn more

- [Edit Linked Documents](#)^[930]

5.6.9.7 Create Linked Document TemplatesTopics

Topic	Detail	See also
Usage	<p>Under the <i>Templates</i> folder, right-click on the Linked Document Templates icon and click on the Create Template context menu option.</p> <p>Enter a name for your template, or select a previously-created template. Click on the OK button.</p> <p>You can group your templates into folders. Right-click on your newly created template and select the Assign Template to Group context menu option. Enter a category name and click on the OK button.</p> <p>You can also modify and delete the templates using the context menu options.</p>	Edit Linked Document Templates ^[934]

Notes

- You can transport these linked document templates between models, using the Export Reference Data and Import Reference Data options

Learn more

- [Export Reference Data](#)^[306]
- [Import Reference Data](#)^[308]






5.6.9.8 Edit Linked Document Templates

Double-click on a previously created template in the Resource View to invoke the Linked Document Template Editor, and right-click on the template to access the editor facilities through the context menu.

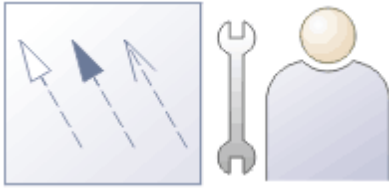
The Document Template Editor is built into Enterprise Architect.

Learn more

- [Scroll Through Text](#)^[2276]
- [File and Print Options](#)^[2277]
- [Cut and Paste Options](#)^[2279]
- [Image and Object Inserts](#)^[2282]
- [Character Formatting](#)^[2284]
- [Paragraph Formatting](#)^[2285]

- [Tab Support](#)  ^[2288]
- [Page Breaks and Repagination](#)  ^[2289]
- [Headers and Footers](#)  ^[2290]
- [Hyperlinks and Bookmarks](#)  ^[2291]
- [Table Commands](#)  ^[2292]
- [Sections and Columns](#)  ^[2296]
- [Stylesheets and Table of Contents](#)  ^[2297]
- [Text/Picture Frame and Drawing Objects](#)  ^[2302]
- [Search/Replace Commands](#)  ^[2303]

5.7 Connectors



UML connectors, along with elements, form the basis of a UML model. Connectors link elements together to denote some kind of logical or functional relationship between them. Each connector has its own purpose, meaning and notation and is used in specific kinds of UML diagrams.

Topics

Topic	Detail	See also
Off-Page Connector	<p>UML, and therefore Enterprise Architect, does not have a connector that continues activity flow between two diagrams.</p> <p>In creating a model diagram, if it becomes necessary to continue flow to another diagram, you should consider revising and simplifying the structure of the process so that groups of Actions are captured in composite Activity elements, and each group of Actions is modeled within the child diagram of an Activity.</p> <p>BPMN, however, does enable you to create off-page connectors.</p> <p>You can also use the Suppress Line Segments menu option to indicate continuation of flow in a large diagram that, when printed, occupies several pages; be aware that these options are purely diagrammatic and do not indicate any diagram relationships in any of the relationship tools.</p>	<p>BPMN ^[156]</p> <p>Style Menu Section ^[940]</p>

Learn more

- [Connector Context Menu](#) ^[936]
- [Connector Tasks](#) ^[942]
- [Connector Properties](#) ^[961]

5.7.1 Connector Context Menu

If you right-click on a connector in a diagram, the connector context menu displays. This provides quick access to some important functions.

Topics

Topic	Detail	See also
Context Menu	<p>The menu is split into up to seven distinct sections:</p> <ul style="list-style-type: none"> • Extensions - displays in the first section only if you have Add- 	<p>Properties Menu Section ^[937]</p> <p>Type-Specific</p>

Topic	Detail	See also
	<p>Ins installed and registered, such as Eclipse</p> <ul style="list-style-type: none"> • Properties • Type Specific • Advanced • Style • Appearance • UML Help - Displays the Enterprise Architect Help topic for this connector type • Delete - delete the connector with this option 	<p>Menu Section ^[938]</p> <p>Advanced Menu Section ^[939]</p> <p>Style Menu Section ^[940]</p> <p>Appearance Menu Section ^[941]</p>
Connector Role Context Menu	<p>For connectors with Roles, right-clicking a connector within up to 60 pixels of an end point displays a role-specific context menu.</p> <p>The Role context menu has three additional menu options:</p> <ul style="list-style-type: none"> • A Source/Target Role... menu option that opens the connector specification dialog with the respective role page selected • A <i>Multiplicity</i> submenu that enables you to set the multiplicity for the role • A Link to Element Feature menu option that displays a dialog through which you can attach the end of the connector to a specific attribute or operation 	<p>Source Role ^[964]</p> <p>Connect To Element Feature ^[944]</p>

Notes

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included

5.7.1.1 Properties Menu Section

The *Properties* section of the connector context menu contains the following options:

Menu Option	Action	Shortcut	See also
<Connector type> Properties	Open the Properties window for the selected connector.		Properties Window ^[961]
Advanced	Display the Advanced menu.		Advanced Menu ^[939]
Attach Note or	Attach a note or constraint to the connector.		Attach a Note or

Menu Option	Action	Shortcut	See also
Constraint			Constraint ^[946]

Notes

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included

5.7.1.2 Type-Specific Menu Section

The *Type-Specific* section of the connector context menu is specific to the object, and only appears for a few different connectors. Some examples are shown below:

Connector	Menu Option	Action	See also
Transition	Locate in State Table	Switch a State Machine diagram to a State Table, with the selected Transition cell highlighted on the table.	
Transition	Message	Set the value of the Message.	
Transition	Find Triggers Associated	Identify hidden triggers, and locate a trigger in the Project Browser; if there: <ul style="list-style-type: none"> Is one trigger, it is immediately highlighted in the Project Browser Are two or more triggers the Element Usage dialog displays, listing the triggers; double-click on the required trigger to highlight it in the Project Browser 	
Transition	Queue Triggers in Simulation	(Available when a simulation is being executed.) Adds the Trigger to the end of the list in the Simulation Events tab of the Breakpoints & Events window, without signalling it.	Simulation Events ^[2128]
Transition	Signal Triggers in Simulation	(Available when a simulation is being executed.) Adds the Trigger to the Simulation Events tab of the Breakpoints & Events window and signals it at the current step of the simulation, leaving it with the status <i>signalled</i> .	Waiting Triggers ^[2129]

Connector	Menu Option	Action	See also
Association	Find Association Class	Locate and highlight the Association Class element in the Project Browser.	
Aggregation	Set Aggregation to Composite	Change the Aggregation to Composite.	
Aggregation	Set Aggregation to Shared	Set the Aggregation to shared; this appears after Set Aggregation to Composite has been selected.	
Information Flow	Find Items Conveyed	<p>(Listed only if there are Information Items on the connector.)</p> <p>Locate the Information Item in the Project Browser; if there:</p> <ul style="list-style-type: none"> • Is one item, it is immediately highlighted in the Project Browser • Are two or more items the Element Usage dialog displays, listing the items; double-click on the required item to highlight it in the Project Browser 	

Notes

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included

5.7.1.3 Advanced Menu Section

The *Advanced* section of the connector context menu contains the following options:

Menu Option	Action	Shortcut	See also
Set Source and Target	Change the source and/or target of the connector.		Change the source and/or target ^[94]
Change Type	Change the connector type.		Change the connector type ^[94]
Change Direction	<p>Keep the ends of the connector the same but change the direction setting. There are four options:</p> <ul style="list-style-type: none"> • Unspecified • Source -> Destination • Destination -> Source 		

Menu Option	Action	Shortcut	See also
	<ul style="list-style-type: none"> • Bi-Directional 		
Reverse Direction	<p>Reverse the ends of the connector; for example, if the connector is an arrow, the arrowhead swaps to the other end.</p> <p>This command maintains the direction of the connector (such as <i>Source -> Destination</i>).</p>		
Specialize Associations	Specify how the properties of this Association specialize the properties of other Associations.		
Information Items Conveyed	Add information items to an Information Flow connector.		Information Item Conveyed ^[1228] Information Flow ^[1225]
Information Flows Realized	Realize any information items conveyed on an Information Flow connector between two elements.		Information Item Realized ^[1229]
Dependency Stereotypes	Select a stereotype for the Dependency (or Trace, Role Binding, Occurrence or Represents connector).		

Notes

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included

5.7.1.4 Style Menu Section

The *Style* section of the connector context menu provides the following options:

Action	Usage	Shortcut	See also
Line Style	<p>Set the connector line style; the options are:</p> <ul style="list-style-type: none"> • Direct • Auto Routing • Custom • Bezier • Tree (Horizontal, Vertical, Lateral Horizontal or Lateral Vertical) 		Set the connector line style ^[948]

Action	Usage	Shortcut	See also
	<ul style="list-style-type: none"> Orthogonal (with Square or Rounded corners) 		
Pin End(s)	<p>Pin the connector start and/or end to the current position on the target element; a sub-menu displays to offer the options of pinning the start point only, the end point only, or both.</p> <p>Once one or both ends are pinned, a fourth option is available to unpin both ends.</p>		
Bend Line at Cursor	Insert an anchor point on the line at the point of the cursor so you can change the shape of the line.	Ctrl + Q	Insert an anchor point ^[948]
Suppress Line Segment	<p>Hide a segment of a connector so that you can view a part of the diagram that it crosses.</p> <p>To reverse the change, right-click on the connector and select the Show All Line Segments context menu option.</p>		
Straighten Line at Cursor	<p>Remove an anchor point on the line at the point of the cursor.</p> <p>(This is the exact opposite of Bend Line at Cursor, and (Ctrl+Q) toggles the connector point between the options).</p>	Ctrl + Q	Connector Styles ^[948]

Notes

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included
- If two connectors cross, the later connector has a 'hoop' at the intersection, indicating that the connectors are crossing

5.7.1.5 Appearance Menu Section

The *Appearance* section of the connector context menu provides the following options:

Menu Option	Action	Shortcut	See also
Appearance	Set the line color and line thickness of the connector.		
Visibility	Set connector visibility; see the table below for sub-menu options.		

Menu Option	Action	Shortcut	See also
Tidy Line Angles	Tidy the line angles of a custom connector.		Tidy the line angles [948]

Visibility Sub-Menu:

Menu Option	Action	Shortcut	See also
Hide Connector	Hide the connector. To show the connector again, follow the steps in the <i>Hide/Show Connectors</i> topic.		Hide/Show Connectors [955]
Hide Connector in Other Diagrams	Hide or show the connector in other diagrams.		Hide/Show Connectors [955]
Hide All Labels	Hide or show all labels attached to the connector.		
Set Label Visibility	Hide or show labels attached to the connector, individually.		Hide or show labels [957]

Notes

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included

5.7.2 Connector Tasks

This topic details some of the tasks associated with managing model connectors, such as:

- [Connect Elements](#) [943]
- [Connect to an Element Feature](#) [944]
- [Change Connector Styles](#) [948]
- [Arrange Connectors](#) [946]
- [Change Connector Type](#) [947]
- [Create Connector in Project Browser](#) [952]
- [Reverse Connector](#) [958]
- [Delete Connectors](#) [954]
- [Hide/Show Connectors](#) [955]

- [Hide/Show Labels](#)^[957]
- [Create Generalization Set](#)^[955]
- [Change the Source or Target Element](#)^[947]
- [Set Relation Visibility](#)^[953]
- [Add a Note to a Connector](#)^[946]
- [Use Tree Style Hierarchy](#)^[960]
- [Create Connector in Project Browser](#)^[952]
- [Show Uses Arrow Head](#)^[959]
- [Set Association Specializations](#)^[958]
- [Change Sequence Message Scope](#)^[959]

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions, if security is enabled, you must have **Update Element** permission to update or delete a connector

Learn more

- [List of Available Permissions](#)^[264]

5.7.2.1 Connect Elements

Topics

Topic	Detail	See also
Connect Elements on a Diagram	<p>The fastest and simplest ways to create connectors are using the Quick Linker and using the Toolbox.</p> <p>To create another connector of the same type as the last connector you used, click on the appropriate source element and press (F3).</p>	<p>Create Connectors In Place Using the Quick Linker^[793]</p> <p>Create Connectors Using the Toolbox^[698]</p> <p>Design Patterns^[1271]</p> <p>Create Domain Specific Connectors From UML Profiles^[1277]</p>
Selecting Connectors	<p>To select a connector, simply click on it; drag handles display, indicating that the connector is selected.</p> <p>This gives the connector focus for keyboard commands such as (Delete), and displays connector properties in docked windows such as the Tagged Values window.</p> <p>If there is more than one connector on a diagram, you can cycle through them using the arrow keys.</p>	
Drag	You can drag a connector to position it; click on the connector	Change the Source or

Topic	Detail	See also
Connectors	and drag it to where it is to appear - note that there are some limitations on how far or to where you can drag a connector. You can also reattach the end of a connector to a different source or target element.	Target Element ^[947]
Connector Properties and Commands	You can double-click on a connector to change properties, or right-click to display the context menu containing commands to change connector type and direction. You can also highlight the connectors on a specific element; select the element and press (L) - all the connectors issuing from or terminating at that element are highlighted.	Connector Properties ^[967] Change Connector Type ^[947] Reverse Connector ^[958]
Create Connectors Without a Diagram	Sometimes it is useful to create relationships between elements without a diagrammatic representation; you can do this using the Project Browser and the Relationship Matrix.	Add Connectors With the Project Browser ^[952] Add Connectors With the Relationship Matrix ^[644]

Notes

- You can reposition a connector by selecting and dragging the connectors as required
- If a connector has source and target roles, you can attach either end of the connector to a specific attribute or operation in the source or target element

Learn more

- [Connect to Element Feature](#)^[944]

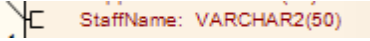
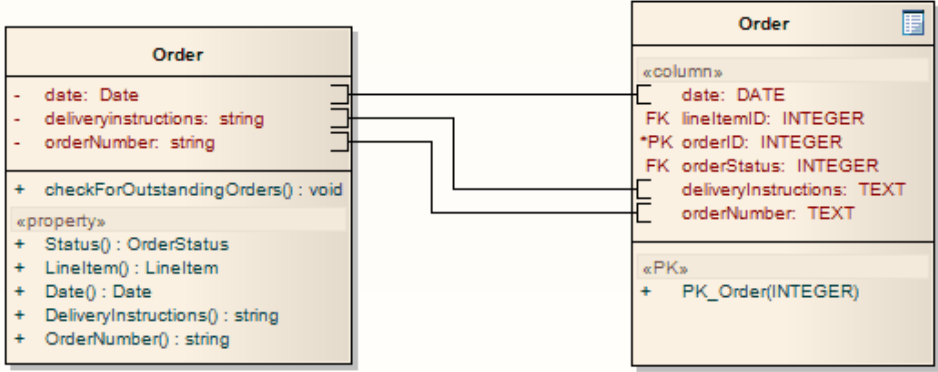
5.7.2.2 Connect to Element Feature

If a connector has source and target roles, you can connect either end of the connector to a specific operation or attribute in the source or target element. This is entirely a visual aid, to indicate which features are significant in the relationship. In code generation or transformation, the link is interpreted as a normal source-element to target-element relationship.

How to

To connect the end of the connector to a specific operation or attribute

Step	Action	See also
1	Right-click on the end of the connector joined to the element containing the required feature.	

Step	Action	See also
2	Select the Link to Element Feature context menu option. The Link to Element Feature dialog displays.	
3	In the Feature Type field, click on the drop-down arrow and select the required feature type - Attribute or Operation . The attributes or operations from the element are listed in the Feature field.	
4	Click on the required attribute or operation, and click on the OK button. The end of the connector changes to a bracket next to the selected feature.  You might create a number of feature-to-feature relationships between two elements (such as a Class and a Table that represents the Class data) to produce a diagram similar to the following:  You can change the feature to which the connector is attached by following the above procedure and selecting the new feature. You can break the link to the selected feature in the following ways: <ul style="list-style-type: none"> • Follow the above procedure, setting the Feature Type to None • Delete the attribute or operation from the element • Change the connector type to a type that does not have source and target roles • Change the connector to a different source or target element that does not contain the feature Note that reversing the direction of the connector does not break the connector's attachment to the feature.	

5.7.2.3 Add a Note to a Connector

This topic describes how you can connect notes and constraints to graphical relationships.

Notes enable you to provide explanations and further detail for one or more connectors on a diagram, with a visible Note element.

How to

To add a note or constraint to a connector

Step	Action	See also
1	Right-click on one of the connectors to attach a note to. The context menu displays.	
2	Select the Attach Note or Constraint menu option. The Link Relations dialog displays.	
3	Select the appropriate radio button in the Link Type panel.	
4	Check all the connectors that participate in the set.	
5	Click on the OK button to complete the note or constraint creation.	
6	Use the normal Note dialog to enter the appropriate text for the note or constraint.	

Notes

- The constraint note is drawn slightly differently to a regular note, and has { and } automatically added to visually indicate the constraint form
- As an alternative to the procedure above, you can drag the Quick Linker arrow on an existing Note or Constraint, and drag it to the required connector; a Notelink is automatically created to attach the Note or Constraint to the connector

Learn more


- [Quick Linker](#)^[79]

5.7.2.4 Arrange Connectors

Connectors between two elements can be moved around the element borders to create a good layout. There is a limit to how much a connector can be moved around, but generally it is very easy to find an acceptable layout. For the best layouts, use the *custom* line style; this enables you to add as many line points and bends as you require to create a clean and readable diagram.

How to

To move and arrange connectors on a diagram

Step	Action	See also
1	Click once on the connector to select it.	
2	Holding the mouse button down, move the connector in the required direction.	
3	To refine the movement, click and hold very near to one end of the connector; this enables a slightly different movement range.	
4	To further refine the movement and range, select either a routed, direct or custom line style; each behaves slightly differently.	Connector Styles 

5.7.2.5 Change Connector Type**How to**

To change a connector type

Step	Action	See also
1	In the Diagram View, right-click on the connector to change. The context menu displays.	
2	Select the Connection Detail Change Type menu option.	
3	In the Connector Type field, click on the drop-down arrow and select the required connector type.	
4	Click on the OK button to apply changes.	

5.7.2.6 Change the Source or Target Element

After you have created a connector between two elements, you might later want to change either the source or target. Instead of deleting and re-creating the connector, Enterprise Architect enables you to change the source or target. There are two ways of doing this: using the Set Source and Target dialog or using the mouse.

How to

To change the source or target element of a connector using the Set Source and Target dialog

Step	Action	See also
1	Right-click on the connector to open the context menu.	
2	Select the Advanced Set Source and Target menu option. The Set Source and Target dialog displays.	
3	Click on the drop-down arrows on the From Element and To Element fields, and select the source and target elements.	
4	Click on the OK button to apply changes.	

To change the source or target element of a connector using the mouse

Step	Action	See also
1	Click on the connector and position the cursor over the 'handle' at one end.	
2	When the cursor changes, click the mouse and drag the handle to the new element.	

Notes

- The connector does not actually move until you release the mouse button over the new source or target element; however:
 - A dotted line shows where the connector would be during the move
 - The solid outline of the nearest element or extension changes to a hatched outline as you move the cursor onto it; this helps you identify where the connector should connect to, if there are many closely-arranged elements, Parts, Ports and other extensions

5.7.2.7 Connector Styles

Connectors come in seven different routing styles. Additionally, if one connector crosses another, the second one 'jumps' over the first (unless the **Enable Connector line jumps** option is deselected on the Options dialog).

Reference

Style	Description	See also
Direct	A straight line from element A to element B. You can move the line (back and forward, up and down) to a limited degree.	
Auto Routing	A vertical and horizontal route from A to B with 90-degree bends. You can move the line to improve the route, but the location and number of bends are not configurable.	
Bezier	A smooth curved line from A to B. Bezier style is directly available for Data Flow diagram connectors, Mind Mapping connectors, State Flows, State Transitions, Object Flows, and Control Flows. <ul style="list-style-type: none"> You can convert other types of relationship to Bezier style by assigning the Tagged Value <i>_Bezier</i>, with an integer value other than 0; however, some relationship types (such as Aggregate) do not accommodate this style very well This Tagged Value over-rides the value of the Style field in the connector Properties dialog 	Connector Properties 1961
Custom Line	The most flexible option; you can add one or more line points and bend and push the line into virtually any shape, using the Toggle Line Point at Cursor option.	
Tree Style - Vertical Tree Style - Horizontal	A line from element A to B with two right-angle bends, and the end points fixed to selected locations on the elements (Vertical or Horizontal). <ul style="list-style-type: none"> You can convert relationships to Tree Style by assigning the Tagged Value <i>_TreeStyle</i>, with a value of H (Horizontal) or V (Vertical) This Tagged Value over-rides the value of the Style field in the connector Properties dialog 	
Lateral - Vertical Lateral - Horizontal	A line from element A to B with a single right-angle bend, and the end points fixed to selected locations on the elements (Vertical or Horizontal). <ul style="list-style-type: none"> You can convert relationships to Lateral style by assigning the Tagged Value <i>_TreeStyle</i>, with a value of LH (lateral-horizontal) or LV (lateral vertical) This Tagged Value over-rides the value of the Style field in the connector Properties dialog 	

Style	Description	See also
Orthogonal - Square Orthogonal - Rounded	You can add one or more line points and bend and push the line into a variety of shapes, using the Toggle Line Point at Cursor option. All line segments thus created are either vertical or horizontal.	

How to

To set the connector style

Step	Action	See also
1	Right-click on the connector to change. The context menu displays.	
2	Select the Line Style option.	
3	From the submenu, select the required style: <ul style="list-style-type: none"> • Direct • Auto Routing • Custom • Tree • Lateral • Orthogonal • (or Bezier, where appropriate) 	

Alternatively:

Step	Action	See also
1	Select the connector to change.	
2	Press the following keys to change the style: <ul style="list-style-type: none"> • (Ctrl+Shift+D) for Direct • (Ctrl+Shift+A) for Auto Routing • (Ctrl+Shift+C) for Custom • (Ctrl+Shift+Z) for Bezier (where appropriate) 	

Step	Action	See also

To bend a connector to quickly and easily route connectors in the required layout

Step	Action	See also
1	Right-click on the connector. The context menu displays.	
2	Set the line style to Custom Line (Ctrl+Shift+C); this enables the Bend Line at Cursor option in the context menu.	
3	Click on the Bend Line at Cursor option to add a line point. Right-clicking a line point displays the Straighten Line at Cursor context menu option, which you can use to remove the line point.	
4	Using the mouse, drag the line point to the required position.	

Alternatively:

Step	Action	See also
1	Hold down (Ctrl) or (Shift) and click on a point on the connector to create a line point. (Ctrl)+ click also removes a line point.	
2	Using the mouse, drag the line point to the required position.	

To tidy line angles (custom connector)

Step	Action	See also
1	Right-click on the connector. The context menu displays.	

Step	Action	See also
2	<p>Click on the Tidy Line Angles menu option; this nudges the custom line in horizontal and vertical increments, saving you the time of trying to get a good layout manually.</p> <p>You can set the Tidy Line Angles option to operate by default; click on the Tools Options menu option to display the Options dialog, and select the Diagram Behavior page.</p>	

To suppress individual line segments

Step	Action	See also
1	<p>Right-click on the connector.</p> <p>The context menu displays.</p>	
2	<p>Set the line style to Custom Line (Ctrl+Shift+C).</p> <p>This enables the Suppress Line Segment option in the context menu.</p>	
3	<p>Click on the Suppress Line Segment option to suppress a line between two bend points.</p> <p>The segment you right-clicked on is suppressed.</p>	
4	<p>To show the segment again, right-click on the line and click on the Show All Line Segments context menu option.</p> <p>One application for this is to represent the continuation of flow when your diagram crosses the page boundary marker in the Diagram View.</p> <p>When you suppress the line segment that crosses the boundary, the link name (connector properties) displays at both ends of the hidden segment; when you print the diagram on multiple pages, the link name identifies the connection apparently broken by the page boundary.</p>	

5.7.2.8 Create Connector in Project Browser

You can create a connector from one element to another directly in the Project Browser.

How to

To connect elements from the Project Browser

Step	Action	See also
1	In the Project Browser, either:	

	<ul style="list-style-type: none"> Right-click on the element to create a connector for, and select the Add Create Link context menu option, or Select the element, press (Insert) and select the Create Link context menu option <p>The Create Link dialog displays.</p>	
2	In the Direction field, click on the drop-down arrow and select the direction of the new connector (Outgoing means this element is the source).	
3	In the Link Type field, click on the drop-down arrow and select the type of connector.	
4	In the Choose target(s) list, click on the name of the target. (If necessary, in the Select Target Type field click on the drop-down arrow and select a feature to list only elements having that feature).	
5	Click on the OK button to create the connector.	

Notes

- You can also reproduce an existing connector between two elements when you paste those elements from the Project Browser into a diagram as instances; an option enables you to copy just the elements, or the relationship as well

Learn more

- [Pasting From the Tree](#)^[733]

5.7.2.9 Relationship Visibility

You can change the visibility of individual connectors or relationships, diagram by diagram.

Access **Diagram | Advanced | Visible Relations (Ctrl+Shift+I)**

How to

To set relationship visibility

Step	Action	See also
1	Open the diagram to change.	

Step	Action	See also
2	Select the Visible Relations menu option. The Set Visible Relations dialog displays.	
3	Select the checkbox against each list item to show, and clear the checkbox against each item to hide. If you want to display the information in a more readable layout, you can resize the dialog.	
4	Click on the OK button to apply the changes.	

5.7.2.10 Delete Connectors

How to

To delete a connector

Step	Action	See also
1	Right-click on the connector and select the Delete Connector context menu option. The Remove Connector dialog displays; this dialog provides the options to hide the connector so that it remains functional, or remove the connector completely.	
2	Click on the appropriate radio button and click on the OK button. If you select the Hide option, it has the same effect as hiding the connector on the Links tab of the source element Properties dialog, or using the Visibility Hide Connector context menu option. It also hides the connector on the Relationships window.	Hiding the Connector ^[955] Relationships ^[648]

Notes

- The dialog does not display if:
 - You have previously selected the **Don't ask again** checkbox or
 - On the Links page of the Options dialog (**Tools | Options | Links**) the **Prompt on connector deletes** checkbox is not selected
- Selecting the **Don't ask again** checkbox also deselects the **Prompt on connector deletes** checkbox
- Selecting the **Prompt on connector deletes** checkbox restores the dialog if you have used the **Don't ask again** checkbox
- If you hide the dialog, the **Delete Connector** context menu option defaults to the setting you last used on the dialog; make sure that you have selected the right option to default to

5.7.2.11 Generalization Sets

A Generalization Set enables you to specify the relationship of a group of Generalizations. Each Generalization is a binary relationship that relates a specific Classifier to a more general Classifier (e.g. from a class to its superclasses). Each Generalization Set defines a particular set of Generalization relationships that describe the way in which a general Classifier (or superclass) can be divided using specific subtypes.

Access [connector context menu](#) | [Advanced](#) | [Generalization Set](#) | [New](#)

How to

To create a generalization set

Step	Action	See also
1	Select the New menu option. The Generalization Set dialog displays.	
2	In the Name field, type the name of the Generalization set; for example, Gender .	
3	In the Power Type field, either: <ul style="list-style-type: none"> Type a new power type, or Click on the drop-down arrow or browser button (...) and select an existing one from the Select <Item> dialog 	Select <Item> Dialog ^[880]
4	Check the IsMember column for the child subtypes that are part of this Generalization set.	

5.7.2.12 Hide/Show Connectors

Connectors/relations that appear in multiple diagrams can be selectively shown or hidden. This makes it easier to read diagrams where elements might have many connectors, but not all are relevant in the context of the current diagram.

How to

To hide or show a connector in the current diagram

Step	Action	See also
1	Double-click on the required diagram element in the Diagram view. The element Properties dialog displays.	

Step	Action	See also
2	Select the Links tab. This lists the connectors linked to the element, whether or not they are hidden on the diagram.	
3	Right-click on the connector to hide or show. The context menu displays.	
4	Select the Show Relation option to show the hidden connector on the diagram, or the Hide Relation option to hide the visible connector. Alternatively, hide a connector by right-clicking on it on the diagram and selecting the Visibility Hide Connector context menu option; however, you must use the Links tab of the element Properties dialog to show the relationship again.	

To hide or show a connector in other diagrams

Step	Action	See also
1	Right-click on the connector in the diagram. The context menu displays.	
2	Select the Visibility Hide Connector in Other Diagrams menu option. The Set Connector Visibility dialog displays.	
3	If the two connected elements have been included in other diagrams, these diagrams are listed here. In the list, all diagrams for which the checkbox is selected show the connector; deselect the checkbox for any diagrams in which to hide the connector. If you want to display the information in a more readable layout, you can resize the dialog. To hide the connector in all of the diagrams listed, click on the Suppress All button.	
4	Click on the OK button to save the changes.	

Notes

- Certain elements, such as Requirements, do not have a Links tab in the Properties dialog

In these cases, open the Relationships window (**View | Relationships**) for the element, right-click on the relationship in the list and select the context menu option to hide or show that relationship in the diagram

Be aware that, in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions with security on, locks on the diagram and elements can make the required option unavailable

Learn more

- [Relationships](#) ^[648]

5.7.2.13 Hide/Show Labels

How to

To hide or display one or more labels on a connector

Step	Action	See also
1	Right-click on the connector. The context menu displays.	
2	Select the Visibility Set Label Visibility menu option. The Label Visibility dialog displays; if you have several long labels, you can resize this dialog for greater clarity.	
3	Select the checkbox against each label to display, and clear the checkbox against each label to hide.	
4	Click on the OK button.	

5.7.2.14 Connector In-place Editing Options

You can edit many of the Enterprise Architect connector labels directly on the diagram. Each label can be bound to a single connector field.

Topic	Detail	See also
Putting a label in Edit Mode	To put a label into Edit mode, either: <ul style="list-style-type: none"> • Select the Edit Label option from the context menu, or • Select a label and press (F2) 	

Topic	Detail	See also
	<p>To save the current text to the field, either press (Return) or deactivate the Edit window</p> <p>To cancel edit mode without saving any changes, press (Esc).</p>	

5.7.2.15 Reverse Connector

You can reverse the direction of a connector without having to delete and re-create it. This is helpful if your design changes or you add the connector wrongly to begin with.

How to

Step	Action	See also
1	Right-click on the incorrect connector.	
2	Select the Connection Detail Reverse Direction context menu option.	

5.7.2.16 Set Association Specializations

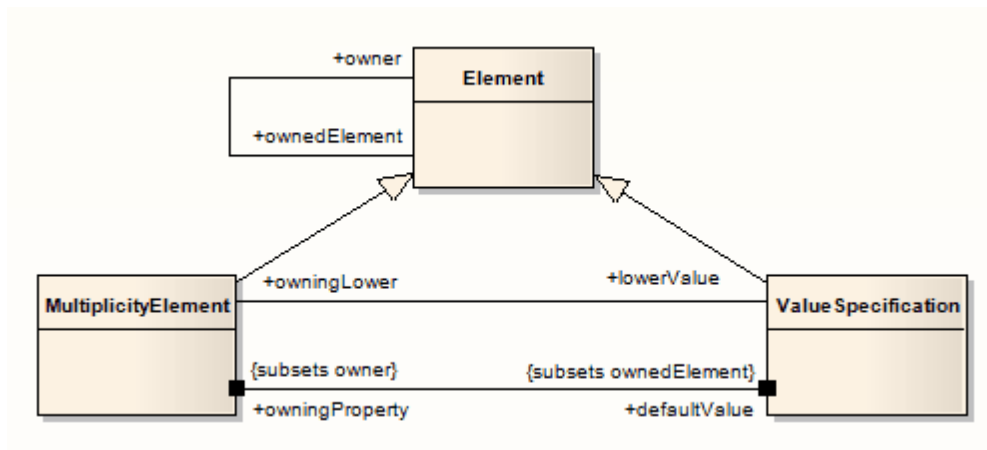
UML enables specialization of properties defined by Associations. Enterprise Architect enables this through the **Specialize Associations** option in the advanced section of the context menu for an Association.

A dialog displays, showing all Associations between the two Classes connected by the current Association and their parents.

The left two columns define the source role of the current Association, while the right two define the target role. With this you are able to select the relationships of each end of the properties listed. When a relationship is set then this is drawn at the corresponding end of the connector on any diagram it appears on.

The dialog displays when you select the **Advanced | Specialize Associations** context menu option on the lowest Association connector in the following diagram.

Example



5.7.2.17 Change Sequence Message Scope

A message in a Sequence diagram represents a dynamic interaction from one element to another. Sometimes when you are designing your model you might have to change either the start or end point of a message as the responsibilities of elements change during design. For this reason, Enterprise Architect enables you to change the message scope by setting a new start or end element.

How to

Step	Action	See also
1	Select the message in the Sequence diagram.	
2	Right-click on the message to open the context menu.	
3	Select Advanced Set Source and Target .	
4	In the pop up dialog, in the From Element and To Element fields, click on the drop-down arrows and select the required elements.	
5	Click on the OK button to save changes. The message is re-routed to meet your changed requirements.	

5.7.2.18 Show Uses Arrow Head

By default the *Use* connector in Use Cases has no arrow head.

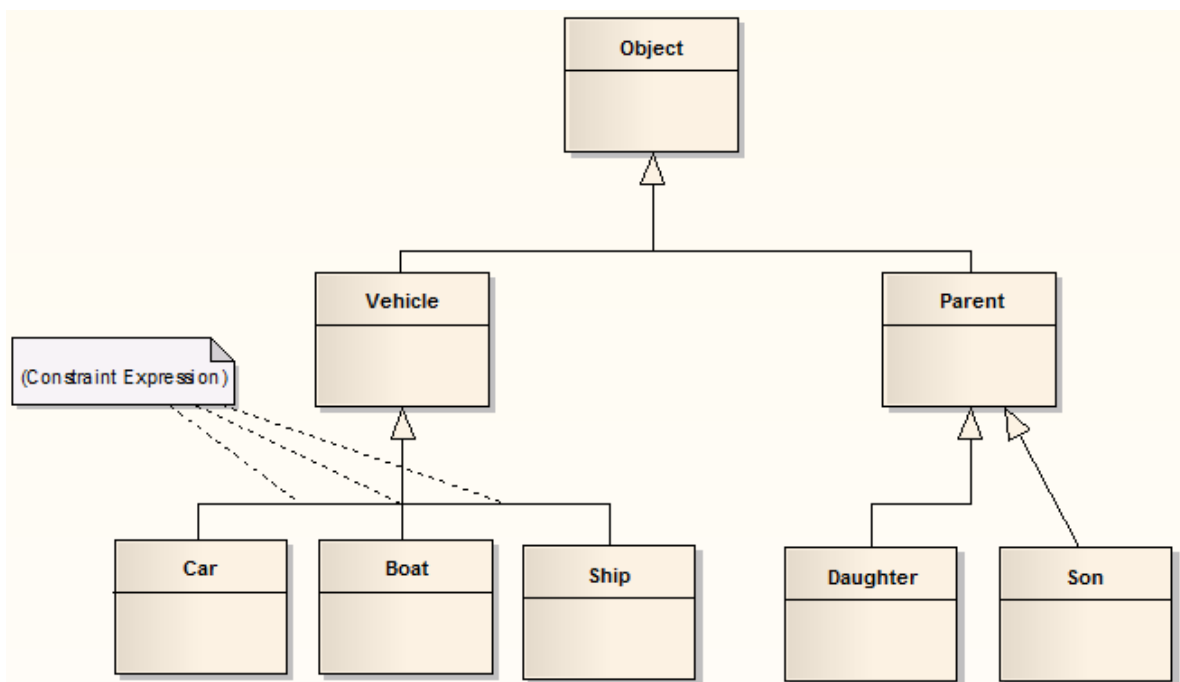
How to

To generate arrow heads on Use connectors

Step	Action	See also
1	Select the Tools Options Links menu option. The Links page of the Options dialog displays.	
2	In the General panel, select the Show Uses arrowheads checkbox.	
3	Click on the Close button. When you save the Use Case diagram, the Use connectors change to display arrowheads.	

5.7.2.19 Tree Style Hierarchy

In Enterprise Architect you can create a tree style inheritance diagram using a special form of the *Generalization* connector, as shown below.



How to

To create a tree style connector

Step	Action	See also
1	Create a normal Generalization between two elements.	
2	Right-click on the connector to open the context menu.	
3	Select the Line Style Tree Style - Vertical or the Line Style Tree Style - Horizontal menu option.	
4	Enterprise Architect automatically makes the Generalization layout conform to a specific shape. By adding more Generalization connectors, and checking their Tree Style options, you can achieve the appearance of the diagram above. You can slide the root and child Classes left and right to achieve the required result; Enterprise Architect maintains the conformity of the branch connectors.	

To set this style of connector as default

Step	Action	See also
1	Select the Tools Options Links menu option. The Links page of the Options dialog displays.	
2	Select the Generalization link style Default = Tree checkbox to make this branching style the default style for inheritance connectors.	

Notes

- In the example diagram, the *Son ->Parent* connector has not yet been put in **Tree Style - Vertical** style
- This style of diagram provides a clearer layout for inheritance hierarchies and is easy to work with

5.7.3 Connector Properties

To access the connector Properties dialog, double-click on a connector in a diagram. You can change several characteristics of connectors from this dialog. Many of these characteristics generate text labels on or around the connector. You can change these labels using the **Label** context menu.

The connector Properties dialog has the following pages:

- General (see below)
- Constraints
- Binding
- Source Role

- Target Role
- Tagged Values
- Advanced properties.

The General page enables you to configure the name of the connector, the direction, the line style, the stereotype (optional) and a comment.

Reference

Field	Usage	See also
Source	Indicates in the name of the source element for the connector.	
Target	Indicates in the name of the target element for the connector.	
Name	(Optional) Indicates a name for the connector; if entered, the name displays on the diagram.	
Alias	(Optional) Indicates an alternative name or alias for the connector.	
Direction	Select the appropriate direction details: from source to destination, destination to source, or bi-directional. Some connectors have arrow heads that depend on this setting; some connectors are logically dependent on this (such as Inheritance).	
Style	Select the appropriate connection style; choose from: <ul style="list-style-type: none"> • Direct • Auto-Routing • Bezier • Custom • Tree (Vertical, Horizontal, Lateral Vertical or Lateral Horizontal), or • Orthogonal (Square Corners or Rounded Corners) 	
Stereotype	(Optional) Indicate the name of a stereotype for the connector, or click on the drop-down arrow and select one. Alternatively, click on the (...) button and select the stereotype from the Stereotype Selector dialog. If entered, the stereotype is displayed on the diagram and overrides the connector type in the RTF documentation.	Stereotype Selector <small>1266</small>

Field	Usage	See also
Virtual Inheritance	Indicates that inheritance is virtual; available only for <i>Generalization</i> connectors.	
Scope	Select the appropriate value for the scope (used for inheritance); available only for <i>Generalization</i> connectors where the child Class is C++.	
Notes	(Optional) Specifies any notes on the connector; the notes are displayed in documentation, if required. As for the Notes window, you can format the text using the Notes toolbar at the top of the field.	Notes Window ^[977] Notes Toolbar ^[979]

Notes

- Generalize, Realize, Associate and Template Binding connectors also have a Binding page, which enables you to define binding expressions and parameter substitutions for the connector

Learn more

- [Message Scope](#)^[959]
- [Constraints](#)^[963]
- [Binding](#)^[964]
- [Source Role](#)^[964]
- [Target Role](#)^[967]
- [Connector Advanced](#)^[968]
- [Tagged Values](#)^[967]
- [Template Binding](#)^[1257]
- [Label](#)^[760]

5.7.3.1 Connector Constraints

A UML connector can also have associated constraints placed on it. Constraints tell us something about the rules and conditions under which a relation operates. For example, it might be a pre-condition that a customer is of a certain type before an Association connector to an Account is allowed.

How to

To set constraints on a connector

Step	Action	See also
1	Double-click on a connector to open the Connector Properties dialog.	

Step	Action	See also
	.	
2	Select the Constraints tab.	
3	Fill in details of the constraint(s) that apply and click on the Save button.	

Reference

Field	Usage	See also
Constraint	The name of the constraint.	
Type	Specifies the type of constraint (such as pre-condition).	
Notes	Indicates notes on the connector.	
Defined Constraints	Displays the list of constraints for this connector.	

Notes

- Constraints about an Association (connector) can be added to further refine the model; constraints detail the business and operational rules for the model

5.7.3.2 Binding

Generalize, Realize, Associate and Template Binding connectors have a Binding tab on their Properties dialog. This enables you to define binding expressions and parameter substitutions for the connector.

Learn more

- [Template Binding](#)^[1257]
- [Parameter Substitution](#)^[1258]
- [Parameterized Classes \(Templates\)](#)^[1183]
- [Class Details](#)^[846]

5.7.3.3 Source Role

A connector can have certain properties assigned to one end, and be associated with the particular role that element plays in the relationship. You can enter details about this role to further develop your model.


This description refers to the role of the *Source* element in a relationship, but applies equally to the role of the *Target* element.

How to

To set the source role details

Step	Action	See also
1	Double-click on a connector to open the Connector Properties dialog.	
2	Select the Source Role tab.	
3	Enter the required details and click on the OK button.	

Reference

Field	Usage	See also
<Type> Role	Enter the name of the role to be played.	
Alias	Enter an alias for the role, if required.	
Role Notes	Enter notes about the role.	
Derived	Indicate that the role value or values can be computed from other information.	
Owned	Indicate that the role is owned by the opposite Class rather than the Association. Selecting this checkbox adds a 'dot' to the appropriate end of the connector ().	
Derived Union	Indicate that the role is derived from the properties that subset it.	
Multiplicity	Specify the role multiplicity (you can define the values of this field on the Cardinality tab of the UML Types dialog). This is the range of instances of the role that can be active in the relationship - for example, <i>one</i> employee can be assigned to tasks; for the target role you define the range of instances (such as tasks) the employee could be assigned to. The values have the following formats: <ul style="list-style-type: none"> • *, or 0..* - zero, one or many instances • 0..n - zero or up to n instances, but no more than n 	Cardinality [886] Attribute properties [889]

Field	Usage	See also
	<ul style="list-style-type: none"> • n - exactly n instances • n..* - n, or more than n instances <p>You can also define source and target element multiplicity in the element Attribute properties.</p>	
Ordered	Indicate that the role is a list and the list is ordered.	
Allow Duplicates	Indicate that the role can contain duplicate elements (relevant only if multiplicity is > 1). This field maps to the UML property <i>isUnique</i> (selecting the checkbox maps to the <i>isUnique</i> value of <i>FALSE</i>).	
Containment	Indicate the nature of the containment at the Destination (reference, value...).	
Access	Indicate the access level for the role.	
Aggregation	Indicate the type of aggregation that this role uses.	
Target Scope	Indicate the level at which this role applies (instance or classifier).	
Navigability	Indicate whether or not this role is navigable (non-navigable ends are shown depending on diagram properties).	
Changeable	Indicate whether this role is subject to change.	
Constraint(s)	Indicate a constraint on the role.	
Qualifier(s)	Indicate any qualifiers or restrictions on the role; separate multiple qualifiers with a semi-colon. Alternatively, click on the (...) button at the end of the field, and define a new qualifier on the Qualifiers dialog (qualifiers typed into the Qualifier (s) field are also automatically added to this dialog).	Qualifiers Dialog ^[1212]
Stereotype	(Optional) Indicate the name of a stereotype that applies to this end of the Association, or click on the (...) button at the end of the field and select a stereotype from the Stereotype Selector dialog.	Stereotype Selector ^[1265]
Member Type	Indicate a role type that can be used when generating collection Classes	

Field	Usage	See also
	for multiplicity > 1.	

Notes

- Source role details are displayed at the start end of a connector; if you have drawn the connector the wrong way, you can always use the **Reverse Direction** menu option from the connector context menu

5.7.3.4 Target Role

A connector can have certain properties assigned to one end, and be associated with the particular role that element can play in the relationship.

You can enter details about this role to further develop your model.

How to

To set the destination role details

Step	Action	See also
1	Double-click on a connector to open the Connector Properties dialog.	
2	Select the Target Role tab.	
3	The details and appearance of this tab are identical to the Source Role tab.	Source Role ⁹⁶⁴

Notes

- Destination role details are displayed at the terminating end of a connector on the diagram

5.7.3.5 Connector Tagged Values

The Tagged Values tab of the connector Properties dialog simply provides the Tagged Values window within the frame of the Properties dialog.

You can define Tagged Values for the connector and, on Association and Aggregation connector types, you can set additional Tagged Values for the source and/or target role.

How to

To set Tagged Values for the connector

Step	Action	See also
1	On the Properties dialog for the connector, click on the Tagged Values tab.	
2	Select the connector type, Connector Source or Connector Target as required.	
3	Either click on the New Tags button or press (Ctrl+N). The Tagged Value dialog displays.	
4	In the Tag field type the tag name and value, or click on the drop-down arrow and select a predefined Tagged Value type.	
5	Click on the OK button to save the changes.	

Learn more

- [Tagged Values](#) ⁹⁶⁹

5.7.3.6 Connector Advanced

The Advanced page displays the advanced properties of the connector (where they exist) and enables you to reset the values of these properties.

How to

To set a value for an advanced (or custom) property


Step	Action	See also
1	Click on the data field to the right of the property name.	
2	Depending on the property, either: <ul style="list-style-type: none"> • Type the value in free text • Click on the drop-down arrow and select the value from the list • Click on the (...) Browse button and search for the required value 	
3	Click on the OK button.	

5.8 Tagged Values



Topics

Topic	Detail	See also
<p>What is a Tagged Value</p>	<p>Tagged Values are a convenient way of adding additional information to an element, beyond what is directly supported by UML. The UML specification provides the Tagged Value element for just this purpose.</p> <p>Often Tagged Values are used during code generation or by other tools to pass information or operate on elements in particular ways. Tagged Values are the preferred method of extending the code generation capabilities of the modeling tool per element / per language.</p> <p>A Tagged Value, strictly, is the value of a property of a modeling item, the property being called a tag. For example, a Class element called <i>Person</i> might have a tag called <i>Age</i> with the Tagged Value of 42. The combination of tag and value, however, is often referred to as a Tagged Value.</p> <p>Enterprise Architect defines a Tagged Value Type, which constrains the possible values of a tag and can specify how a value is assigned to the tag. For example, the tag <i>Age</i> might have a Tagged Value Type of <i>Integer</i>, so the user simply types in a numeric value. Alternatively, the type could be <i>Spin</i>, with lower and upper limits of, say, 20 and 120, so the user sets a value by clicking on arrows in the field to increment or decrement the value within the limits of 20 and 120.</p> <p>To quickly add Tagged Values to one of more elements, see the <i>Quick Add - Tagged Value to Elements</i> topic.</p>	<p>Tagged Value Type ^[985]</p> <p>Quick Add - Tagged Value to Elements ^[971]</p>
<p>The Tagged Values Window</p>	<p>The Tagged Values window is used to view and modify Tagged Values for the currently selected modeling item, either in the current diagram or in the Project Browser.</p> <p>The Tagged Values window is a dockable window. You can use it to perform the following actions:</p> <ul style="list-style-type: none"> • Assign a Tagged Value to an Item • Modify Tagged Values • Assign Notes to a Tagged Value • Perform advanced tag management <p>A Technology Developer can also create new structured Tagged Values, reference data Tagged Values and custom Tagged Values from predefined Tagged Value Types.</p>	<p>Assign a Tagged Value to an Item ^[972]</p> <p>Modify Tagged Values ^[973]</p> <p>Assign Notes to a Tagged Value ^[973]</p> <p>Perform Advanced Tag Management ^[975]</p> <p>Custom Tagged Values ^[1393]</p> <p>Tagged Value Types ^[1385]</p>

Topic	Detail	See also														
<p>Model Elements and Features with Tagged Values</p>	<p>The following model components can use the Tagged Values window as a convenient way to quickly view and modify Tagged Values:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #e0e0e0;">Component</th> <th style="background-color: #e0e0e0;">Description</th> </tr> </thead> <tbody> <tr> <td>Elements</td> <td>Elements display their own Tagged Values and inherited values.</td> </tr> <tr> <td>Object Instances</td> <td>Object Instances display owned tags and those inherited from their classifier.</td> </tr> <tr> <td>Ports and Parts</td> <td>Ports and parts display information similar to Port/Part 'Type' instead of a classifier. Tags are included for all parents and other siblings of the same type.</td> </tr> <tr> <td>Attributes</td> <td>Include owned Tagged Values and those inherited from type classifiers, with the inclusion of any inherited values.</td> </tr> <tr> <td>Operations</td> <td>Owned properties only.</td> </tr> <tr> <td>Connectors</td> <td>Owned properties only.</td> </tr> </tbody> </table> <p>When over-riding an inherited property, Enterprise Architect copies the tag from the parent down to the child element and sets the new value, leaving the original tag unchanged.</p>	Component	Description	Elements	Elements display their own Tagged Values and inherited values.	Object Instances	Object Instances display owned tags and those inherited from their classifier.	Ports and Parts	Ports and parts display information similar to Port/Part 'Type' instead of a classifier. Tags are included for all parents and other siblings of the same type.	Attributes	Include owned Tagged Values and those inherited from type classifiers, with the inclusion of any inherited values.	Operations	Owned properties only.	Connectors	Owned properties only.	
Component	Description															
Elements	Elements display their own Tagged Values and inherited values.															
Object Instances	Object Instances display owned tags and those inherited from their classifier.															
Ports and Parts	Ports and parts display information similar to Port/Part 'Type' instead of a classifier. Tags are included for all parents and other siblings of the same type.															
Attributes	Include owned Tagged Values and those inherited from type classifiers, with the inclusion of any inherited values.															
Operations	Owned properties only.															
Connectors	Owned properties only.															
<p>Tagged Values Toolbar Buttons</p>	<p>The buttons in the Tagged Values toolbar enable you to add, edit, sort, delete and arrange the Tagged Values of model features</p>  <p>From left to right, the button functions are as follows:</p> <ul style="list-style-type: none"> • The Show in compartments button displays the Tagged Values in element compartments on diagrams • The Sort and Show Alphabetically button sorts the current Tagged Values for the element alphabetically • The New Tagged Value button adds a new tag, to which you assign a value 															

Topic	Detail	See also
	<ul style="list-style-type: none"> • The Edit Tagged Value Notes button enables you to create notes that explain the purpose of the Tagged Value • The Delete Tagged Value button removes the currently selected Tagged Value • The Default Tagged Value Types button enables quick access to tag definitions created in the Configuration menu • The Tagged Value Options button enables you to show or hide the fully qualified paths for the Tagged Values in the window, and to show duplicate Tagged Values • The Help button displays help relating to use of the Tagged Values window 	

5.8.1 Quick Add - Tagged Value To Elements

You can add a Tagged Value to one or more elements with a special shortcut. You can also use the Current Element toolbar. The last button is a shortcut to the **Add Tagged Value** function.

To delete this property you must open the element Properties dialog, go to the Tagged Values tab and manually delete the item. There is currently no shortcut to delete tags from multiple elements simultaneously.

To add notes to the Tagged Value, go to the Tagged Values tab, click on the Tagged Value name, and click on the **Edit Notes** button in the tab toolbar. The Notes dialog displays.

Any Notes text you enter also displays in the **Info** section at the bottom of the Tagged Values window.

Access **Add | Tagged Value**

Use to

- Add Tagged values to elements

How to

Step	Action	See also
1	<p>From an element context menu (or the context menu of a multi-selection) choose the Add Tagged Value menu option.</p> <p>(Alternatively, select one or more elements and press (Shift+Ctrl+T).)</p> <p>The Tagged Values dialog displays, which enables you to enter a Name and Value for the tag.</p>	
2	<p>Click on the OK button to add your new Tagged Value to all the currently selected elements.</p>	

Step	Action	See also

5.8.2 Assign a Tagged Value to an Item


You can assign Tagged Values to several model features.

Access [View | Tagged Value](#)

Use to

- Assign a Tagged Value to an item

How to

Step	Action	See also
1	Select the model feature to associate with the defined Tagged Value.	
2	Ensure that the Tagged Values window is visible (select the View Tagged Values menu option, or press (Ctrl+Shift+6)). Alternatively, open the Properties dialog for the object and select the Tagged Values tab.	
3	Either click on the New Tags button  or press (Ctrl+N). The Tagged Value dialog displays.	
4	In the Tag field, type the tag name or click on the drop-down arrow and select the appropriate tag to assign to the item.	
5	If appropriate, type a specific value for the tag in the Value field.	
6	To confirm selection of the Tagged Value, click on the OK button.	

Learn more

- [Predefined Tagged Value Types](#) ^[1385]
- [Assign Notes to a Tagged Value](#) ^[973]
- [Modify Tagged Values](#) ^[973]
- [Model Elements and Features with Tagged Values](#) ^[970]

5.8.3 Assign Notes to a Tagged Value

Once a Tagged Value has been assigned to a model feature, you can add information and notes describing the Tagged Value to the information property of the Tagged Value.

Access [View | Tagged Values](#) (Ctrl+Shift+6)

Use to

- Assign a note to a Tagged Value

How to

Step	Action	See also
1	Select the Tagged Values menu option. The Tagged Values window displays.	
2	Click on the model feature for which to edit the Tagged Values. Its Tagged Values display in the Tagged Values window.	
3	Click on the Tagged Value to add information to.	
4	Click on the Edit Tagged Value Notes button or press (Ctrl+E). The Tagged Value Note dialog displays.	
5	In the Note field, type the information relating to the Tagged Value. Click on the OK button. The information is displayed in the lower portion of the Tagged Values window whenever the Tagged Value is selected.	

Learn more

- [Model Features](#)^[970]

5.8.4 Modify Tagged Values

Once a Tagged Value has been assigned to the model feature it is possible to edit the values from the Tagged Values window.

Use to

- Edit Tagged Values

How to

Step	Action	See also
1	Click on the View Tagged Values menu option, or press (Ctrl+Shift+6). The Tagged Values window displays.	
2	Click on the model feature for which to edit the Tagged Values. The window shows all of the tags for the selected feature, each with their current value.	
3	Edit the fields as appropriate; the information entered can only reflect the value types that have been defined by the tag's Tagged Value Type. There are four types of value field for a Tagged Value: <ul style="list-style-type: none"> • 'Open' fields', in which you can type any appropriate value • 'Drop-down list' fields, where you click on the drop-down arrow to select from a discrete list of possible values such as M or F, or Win, Lose or Draw • 'Spin' fields, where you click on up or down arrows in the field to increase or decrease the value within certain limits • 'Further detail' fields, where you click on an ellipsis (...) in the field to display a dialog in which you enter information (such as notes) or indicate a source of further information (such as a classifier) 	Instance Classifier 880

Notes

- To override a Tagged Value defined in a parent element, edit the value in the from <parentname> compartment of the Tagged Values window

Once this has been done the tag is moved into the selected element's Tagged Values; this does not affect the Tagged Values defined in the parent element

5.8.5 Show Duplicate Tags

Tagged Values are by default set to hide duplicate values. This setting is used to facilitate inherited and overridden tag names.

How to

To set the Tagged Values window to show duplicate values

Step	Action	See also
1	On the Tagged Values window toolbar, click on the Options icon.	
2	Select the Show Duplicate Tags context menu option.	

Alternatively:

Step	Action	See also
1	Select the Tools Options menu option. The Options dialog displays.	
2	From the hierarchical tree, select the Objects item.	
3	Select the Show Duplicate Tags checkbox.	

Notes

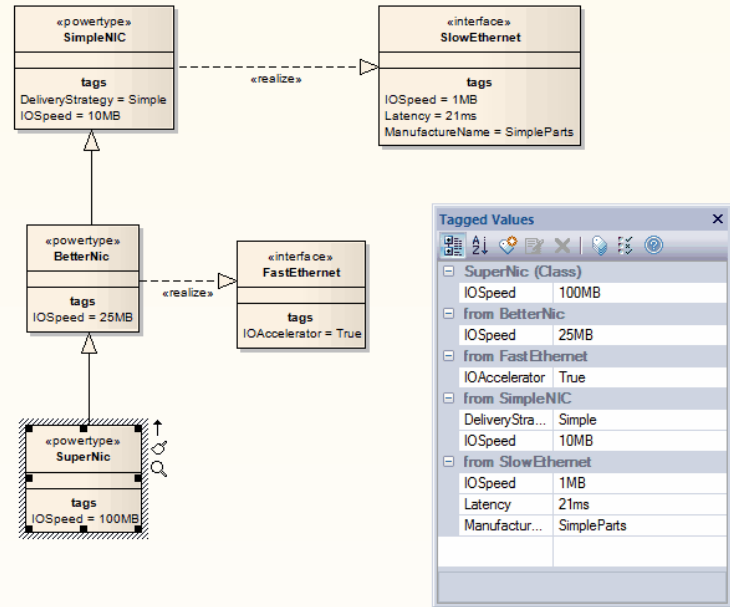
- In either procedure, to hide duplicate values again deselect the option or check box

5.8.6 Advanced Tag Management

Tagged Values can also be managed within a type hierarchy and with respect to element instances, using the Tagged Values window.

Topics

Topic	Detail	See also
Usage	Using the Tagged Values window it is possible to: <ul style="list-style-type: none"> • View Tagged Values inherited from parent Classes or realized interfaces or applied stereotypes • Override Tagged Values derived from parents or applied stereotypes with a unique value for the current element • Delete Tagged Values from the current element (if a parent version of the Tagged Value exists, it re-appears in the list after the override is deleted) 	
Tag Hierarchy	The diagram below illustrates a complex tag hierarchy and the way Tagged Values can be either inherited or overridden in specialized Classes to create the final tagged property set for an element.	

Topic	Detail	See also
	<p>Note also that a similar concept applies to instances, in which case the full tag set is created from the directly owned tags, plus all of those merged in from the classifier's type hierarchy, additional stereotypes and realized interfaces.</p> 	

Learn more

- [Tagged Values](#) 969

5.9 Notes

Notes are the main documentation feature you use to describe an element, diagram, feature or connector; in the documentation that Enterprise Architect generates, notes feature prominently. You can record and view notes on a modeling object using the Notes window.

Access **View | Notes** (Ctrl + Shift + 1)

Topics

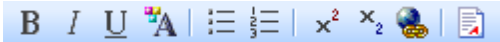
Topic	Detail	See also
Usage	<p>You use the Notes window to view and edit the documentation (notes) associated with elements, diagrams, attributes, operations and connectors, either from a diagram or from the Project Browser.</p> <p>When you select an element, the note displayed changes to reflect the current selection; if you make changes to notes in this window, they are saved.</p> <p>If you want to display the Notes information in a more readable layout, you can resize the window.</p> <p>You can also format the notes text using the Notes toolbar at the top of the Notes window.</p> <p>You can cut, copy, paste and delete text in the Notes window, or in any Notes or Description field that shows the Notes toolbar, using a right-click context menu; from the context menu, you can also select an option to spell-check a highlighted word.</p> <p>You can also edit notes by double-clicking on an element or connector in a diagram or in the Project Browser, to open the Properties dialog; any formatting changes made in one display are reflected in the other.</p> <p>On the Testing, Maintenance and Project Management windows, any descriptive, history, input or results text for a selected item is also displayed in the Notes window; you cannot edit this text in the Notes window.</p>	Notes Toolbar ⁹⁷⁹
Glossary Entries	<p>The Notes window or field also enables you to create a Project Glossary entry from text you have highlighted in the window or field.</p> <p>To create the Glossary entry</p> <ul style="list-style-type: none"> • Highlight the notes text to use as the Glossary definition (if suitable text is available), and press (Ctrl+C) to copy it • Highlight the text to use as the Glossary term, and right-click on it to display the context menu • Select the Create Glossary Definition menu option; the Glossary Detail dialog displays, with the selected term in the Term field 	Project Glossary ⁴⁶⁷ Glossary Detail ⁴⁶⁸

Topic	Detail	See also
	<ul style="list-style-type: none"> • If you have copied some definition text, paste it into the Meaning field; otherwise type a suitable definition of the term in this field • In the Type field, select the appropriate term type • Click on the Apply button to save the new Glossary definition <p>The term displays in the Notes text as a roll-over hyperlink which, when you move the cursor over it, displays the Glossary definition of the term.</p> <p>Having created a glossary definition anywhere else in the model, you can insert the glossary term in the text of the Notes window (or Notes panel of a dialog) as a rollover hyperlink to the definition.</p>	
Inserting a Glossary term in a Notes Window	<ul style="list-style-type: none"> • In the Notes window, move the cursor to the point in the text at which to insert the glossary term • Press (Ctrl+Space); the glossary term selection list displays • Double-click on the term to insert in the Notes text; the term is inserted as a rollover hyperlink to the definition 	
Additional keyboard shortcuts	<ul style="list-style-type: none"> • Insert Date and Time (F5) • Select line of text (F8) • Spell check Notes text only (F7) • Undo changes (Ctrl+Z) • Redo changes (Ctrl+Y) or (Ctrl+Shift+Z) • Copy (Ctrl+C) • Paste (Ctrl+V) • Cut (Ctrl+X) <p>Any Note text appearing in the element Note compartments in diagrams is not formatted.</p>	

5.9.1 Notes Toolbar

Although it is not an independent toolbar that you can pin to the screen top or sides, or float in your work area, the Notes toolbar appears in many places across Enterprise Architect in the **Notes**, **Description** and **History** fields.

Topics

Topic	Detail	See also
Where the Toolbar Appears	Throughout Enterprise Architect on various tabs and pages of, for example, the: <ul style="list-style-type: none"> • Element and Connector Properties dialogs • Operations and Attributes Properties dialogs • Diagram Properties dialog • Hyperlink Details dialog • Testing window descriptions • Notes window • Glossary Detail dialog • Assigned Resources dialog • Project Management window • Scenarios & Requirements window 	
Options	 <p>The options of this toolbar operate on selected text and any new text continuing from the formatting.</p> <p>The options (with some keyboard shortcuts) are, from left to right:</p> <ul style="list-style-type: none"> • Make text bold (Ctrl+B) • Make text italic (Ctrl+I) • Underline text (Ctrl+U) • Change the font color of the text • Insert list bullet points (Ctrl+.) (full stop) • Insert list numbering (Ctrl+1) • Make text superscript • Make text subscript • Insert a hyperlink - this displays the Hyperlink Details dialog, on which you specify the type of hyperlink and type in or browse for the location of the target of the hyperlink; you can use full paths or local (path substitution) paths 	Hyperlinks <small>[1668]</small> Linked Documents <small>[929]</small>

	<ul style="list-style-type: none">• Create a new linked document for the element, or edit an existing one; if the element does not have a linked document, the New Linked Document dialog displays	
--	--	--

Notes

- If the toolbar is displayed but grayed out, the text field is read-only and cannot be edited; other long text fields in Enterprise Architect might not have the toolbar, in which case the Notes facility is not available for those fields
- For any Notes text that is displayed on a diagram, you must select the **Render Formatted Notes** checkbox on the Feature Visibility dialog in order to reproduce the formatting
- You can create a Project Glossary term and definition from text in any field that has the Notes toolbar

Learn more

- [Feature Visibility dialog](#)^[745]
- [Project Glossary](#)^[467]
- [Notes](#)^[977]

5.10 Reference Data



Reference data is used in many places to provide content for drop-down list boxes. Setting up a project often involves setting up the base set of reference types to use. Reference data options can be set up from the **Settings** menu, including:

- People
- General Types
- Maintenance
- Metrics and Estimation
- UML
- Data Types

Having set up the reference data in a project, you can also export and import it between projects.

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Manage Reference Data - Update permission to update and delete reference items.

Learn more

- [People](#)^[989]
- [General Types](#)^[995]
- [Maintenance](#)^[1001]
- [Metrics and Estimation](#)^[1000]
- [UML](#)^[981]
- [Data Types](#)^[987]
- [Export and Import](#)^[305]
- [Manage Reference Data - Update](#)^[264]
- [Applying Stereotypes](#)^[1265]
- [Custom Stereotypes](#)^[1293]

5.10.1 UML Types

The UML Types dialog enables you to configure stereotypes, Tagged Value types and the cardinality list for your project.

Access [Settings | UML Types](#)

Use to

- Add, modify and delete stereotypes
- Customize stereotypes appearance
- Updating shape scripts and metafiles associated with Stereotypes
- Create/modify tagged values
- Define cardinality rules

Learn more

- [Stereotypes](#)^[982]
- [Tagged Value Types](#)^[985]
- [Cardinality](#)^[986]

5.10.1.1 Stereotype Settings

Enterprise Architect has an extensive set of Standard Element Stereotypes that you can apply to any UML construct. Using the Stereotypes tab of the UML Types dialog, a Technical Developer can also customize the stereotypes for your project by adding, modifying and deleting them. For information on customizing stereotypes, see *Custom Stereotypes*.

Stereotypes can be modified to make use of metafiles (image files) or customized colors, or to make use of the Enterprise Architect Shape Scripts to make new element shapes to determine the shape and dimensions of the element.


Access [Settings | UML Types > Stereotypes](#)

Use to

- Add, modify and delete stereotypes
- Customize stereotypes appearance
- Updating shape scripts and metafiles associated with stereotypes

Reference

Field	Usage	See also
Stereotype	Specify the name of the stereotype.	
Group name	Enable grouping of stereotype features by a plural name, for attributes and operations, which is shown on diagrams in the attribute and operations compartments.	
Base Class	Enable the stereotyped element to inherit the base characteristics from a pre-existing element type.	

Field	Usage	See also
Notes	Indicates any notes concerning the stereotype (not the elements to which the stereotype is to be applied).	
None	Switch to the default element appearance.	
Metafile	Enable an image file to be used for the appearance of the stereotype.	
Shape Script	Specify custom shapes for the stereotype using the Enterprise Architect Shape Scripting language. For more information see the Shape Scripts topic.	Shape Scripts 
Assign	Add the associated metafile or Shape Script from the stereotyped element.	
Remove	Remove the associated metafile or Shape Script from the stereotyped element.	
Fill	Set the default background color of the element.	
Border	Control the border color.	
Font	Control the color of the stereotype font.	
Reset	Reset the appearance of the element to the default element appearance.	
Preview	Provides a visual representation of the currently active Stereotype.	
New	Enables a new stereotype to be created.	
Save	Saves the Stereotype details and adds the Stereotype to the Defined Stereotypes list.	
Delete	Removes a Stereotype from the Defined Stereotypes list.	

Learn more

- [Standard Element Stereotypes](#) ^[1267]
- [Applying Stereotypes](#) ^[1265]
- [Custom Stereotypes](#) ^[1293]

5.10.1.1.1 Shape Editor

The Shape Editor enables a Technology Developer to specify custom shapes via a scripting language; that is, to create *Shape Scripts*. These custom shapes are drawn instead of the standard UML notation. Each script is associated with a particular Stereotype, and is drawn for every element of that stereotype.

Access [Settings | UML Types > Stereotypes : Assign](#)

Use to

- Create a custom shape

Reference

Field	Usage	See also
Format	The format in which the script is written.	
Import	Opens a Windows browser dialog, allowing for a script file to be loaded.	
Export	Opens a Windows browser dialog, allowing for a script file to be saved.	
Edit window	The area in which a script can be written.	
Preview of main	A visual representation of the compiled script. Nothing displays until the Refresh button is clicked.	
Next Shape	If a composite shape is defined within the Edit Window, clicking on the Next Shape button pages through the components of the shape.	
Refresh	Parses your script and produces a visual example of the shape in the Preview of main window.	
OK	Exit from the Shape Editor; don't forget to save your scripts from the Stereotype tab.	Getting Started With Shape Scripts ^[1358]

--	--	--

Notes

- Shape Scripts adopt the same color gradient settings as normal elements, as defined in the Standard Colors page of the Options dialog
- If an element's appearance is modified by a Shape Script, many of the **Advanced** context menu options for that element are disabled
- Once you have finishing writing your Shape Script, click on the **OK** button; to save the Shape Script you must click on the Stereotypes tab

Learn more

- [Writing Scripts](#) ^[1363]
- [Advanced Settings](#) ^[829]

5.10.1.2 Tagged Value Types

Tagged Values are used in a variety of places within Enterprise Architect to specify additional information about an element or connector. The Tagged Value Types tab of the UML Types dialog enables a Technology Developer to rapidly create Tagged Values, using a range of predefined structured Tagged Values to create structured tags that adhere to a specific format. For example, for model features that use the predefined tag Boolean you can use the Tagged Values window to assign a value of *True* or *False* and no other value.

You can also add default Tagged Value names and create predefined reference data Tagged Value types and custom masked Tagged Value types.

Any Tagged Value names created display in the drop-down lists of Tagged Value names in the Tagged Value dialogs for elements, operations and attributes.

Access **Settings | UML Types > Tagged Value Types**

Use to

- Create new tagged values

Reference

Field	Usage	See also
Tag Name	Type the new name of the Tagged Value.	
Description	Type a description of the Tagged Value.	
Detail	Type any additional information necessary.	

Field	Usage	See also
New	Click to begin creating a new Tagged Value.	
Save	Click to save the details of a Tagged Value.	
Delete	Click to delete a Tagged Value from the Defined Tag Types list.	
Defined Tag Types	A list of previously defined Tagged Values.	

Notes

- You can transport these Tagged Value Type definitions between models, using the Export Reference Data and Import Reference Data options; Tagged Value Types are exported as *Property Types*

Learn more

- [Predefined Structured Tagged Values](#) ^[1386]
- [Create Structured Tags](#) ^[1389]
- [Predefined Reference Data Tagged Value Types](#) ^[1392]
- [Custom Masked Tagged Value Types](#) ^[1393]
- [Tagged Values Window](#) ^[969]
- [Export Reference Data](#) ^[306]
- [Import Reference Data](#) ^[308]

5.10.1.3 Cardinality

The Cardinality Values tab of the UML Types dialog enables you to add, modify and delete values in the default cardinality list.

The cardinality values are used to define the multiplicity of source and target elements in relationships. This is the range of instances of the role that can be active in the relationship; for example, one employee can be assigned to tasks; for the target role you define the range of instances (such as tasks) the employee could be assigned to.

The cardinality values are also used to define the multiplicity of a Classifying element; that is, the number of instances of the element that can exist. For example, the Class element *Building Walls* might have a multiplicity of 2..n, meaning that at least two walls must exist (to support the roof) but there can be many walls if the building design required it.

The values have the following formats:

- ***, or **0..*** - zero, one or many instances
- 0..n** - zero or up to n instances, but no more than n
- n** - exactly n instances

- **n..*** - n, or more than n instances.

Access [Settings | UML Types > Cardinality Values](#)

Use to

- Create Cardinality values

Reference

Field	Usage	See also
Cardinality	Type the new name of the Cardinality Value.	
New	Type a description of the Cardinality Value.	
Save	Saves the Cardinality value and adds it to the Cardinality List.	
Delete	Deletes a Cardinality value from the Cardinality List.	
Cardinality List	A list of already defined Cardinality values.	

Notes

- You can transport these cardinality values between models, using the Export Reference Data and Import Reference Data options

Learn more

- [Export Reference Data](#)^[306]
- [Import Reference Data](#)^[308]
- [Source Role](#)^[964]
- [Destination Role](#)^[967]

5.10.1.4 Data Types

Different programming languages support different inbuilt data types. The **Programming Languages Datatypes** dialog enables you to extend and manage the set of inbuilt data types associated with a language as well as create new programming languages for use within Enterprise Architect.

Access [Settings > Code Datatypes](#)

Use to

- Create new programming languages to use within Enterprise Architect
- Extend and manage inbuilt data types

Reference

Field	Usage	See also
Product Name	Specify the name of the programming language.	
Add Product	Add a new programming language to the drop-down fields for Class elements within the Enterprise Architect model and enable the new language to be made available to the Code Template Editor once at least one datatype has been added to the language.	Code Template Editor ^[1404]
Datatype	Specify the name of the datatype; this is the language-specific name of the datatype.	
Common Type	Specify the common type, the generic name of the datatype; for example, the Java <i>boolean</i> datatype has a common datatype <i>Boolean</i> .	
New	Create a new data type.	
Save	Save the newly created datatype.	
Delete	Delete the selected datatype.	

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Configure Datatypes** permission to update and delete data types
- You can delete data types that you have defined, but you cannot delete any of the predefined data types
- You can transport these data types between models, using the Export Reference Data and Import Reference Data options

Learn more

- [Export Reference Data](#)^[306]
- [Import Reference Data](#)^[308]

- [Permission List](#)^[964]

5.10.2 People

Access [Settings](#) | [Project Types](#) | [People](#)

Use to

- Maintain Authors involved within a project
- Define role types that are captured within Enterprise Architect
- Record information on project resources
- Capture client details associated with the current model

Learn more

- [Project Authors](#)^[989]
- [Project Roles](#)^[991]
- [Project Resources](#)^[992]
- [Project Clients](#)^[993]

5.10.2.1 Project Authors

You can define the people who are working on a project, such as the authors of specific elements.

Access [Settings](#) | [Project Types](#) | [People](#) > [Project Authors\(s\)](#)

Use to

- Maintain Authors involved within a project

Reference

Field	Usage	See also
Name(s)	<p>Type the name of the person registered as a Project Author.</p> <p>If you are using a Windows Active Directory, you can select names from the directory; click on the (...) (Browse) button to display the Select Users dialog.</p> <p>You can also type a list of names separated by semi-colons; this enables you to define a group of people sharing a role, such as a team of Developers, Testers or Analysts.</p> <p>Do not leave any spaces between the names and the semicolons.</p>	<p>Select Users^[990]</p>

Role	Enter the role the Project Author plays in the project (such as Designer, Analyst, or Architect). You can type a role name or click on the drop-down arrow and select a role defined through the Project Roles tab.	Project Roles ⁹⁹¹
Notes	Type any additional notes concerning the Project Author.	
New	Add further Authors.	
Save	Add the new Author to the Defined Authors list.	
Delete	Removes an Author from the Defined Authors list.	
Defined Authors	Review the Project Authors already defined.	

Notes

- If you enter multiple names, Enterprise Architect adds them separately and in alphabetical order to the **Defined Authors** list; if you then click on one of these names, Enterprise Architect displays that name only in the **Name(s)** field
- If you type in a role, this is not added to the roles on the Project Roles tab
- You can transport these author definitions between models, using the Export Reference Data and Import Reference Data options

Learn more

- [Export Reference Data](#) ³⁰⁶
- [Import Reference Data](#) ³⁰⁸

5.10.2.1.1 Select Users

If your company is using a Windows Active Directory, you can select the Project Author names from the local or corporate-wide directory.

Access [Settings | Project Types | People > Project Authors\(s\) : \(...\)](#)

Use to

- Select Project Authors from Windows Active Directory

Reference

Field	Usage	See also
Object Types...	Opens the Object Types dialog, which provides a choice of object types that can be used.	
Locations...	Defines the root location from which to begin a search.	
Check Names	Matches object names listed in the Enter the object names to select section.	
Enter the object names to select	Specify object names to search for.	
Advanced...	Opens the Advanced dialog to provide further search options. For more information on this dialog, please refer to Windows Help and Support.	

Notes

- Multiple Entries can be typed into the **Enter the object names to select** section; ensure that objects are separated by semicolons (for example: Name01; Name02)

5.10.2.2 Project Roles

People associated with a project play a *role* in analysis, design or implementation, such as Application Analyst, Architect, Developer and Project Manager. Project roles define the activities that resources can undertake.

Access **Settings | Project Types | People > Project Roles**

Use to

- Define role types that are captured within Enterprise Architect

Reference

Field	Usage	See also
Role	Type or select the name of the role.	
Description	Type a description of the role .	
Notes	Type any additional information related to the role.	

New	Add further Roles.	
Save	Add the new role to the Defined Roles list.	
Delete	Removes a role from the Defined Role list.	
Defined Roles	<p>Review the Project Roles already defined.</p> <p>The Defined Roles list is available for selection for any element in the model; for example, you can select roles on the Project Authors tab of the People dialog, and the Resource Allocation tab of the Project Management window.</p> <p>You can also specify other roles on these dialogs, but such roles are not added to the Defined Roles list.</p>	<p>Project Authors ^[989]</p> <p>Resource Allocation ^[450]</p>

Notes

- Deleting a role has no effect on any Project Author definition having this role; the deleted role becomes a simple text entry in the Project Author definition
- You can transport these role definitions between models, using the **Export Reference Data** and **Import Reference Data** options

Learn more

- [Export Reference Data](#) ^[306]
- [Import Reference Data](#) ^[308]

5.10.2.3 Project Resources

Resources are, for example, project authors, analysts, programmers and architects. That is, anyone who might work on the system over time, either adding to the model or programming and designing elements of the system outside Enterprise Architect.

Access **Settings | Project Types | People > Project Resources**

Use to

- Record information on project resources

Reference

Field	Usage	See also
Name	Type or select the name of the person listed as a resource. The resource name is available for use in Resource Management.	Resource Management ^[450]
Organization	Type the name of the organization employing the resource.	
Role(s)	Type or select the role the resource plays in the project (for example, Designer, Analyst, Architect).	
Phone 1, Phone 2, Mobile, Fax	Type the contact telephone numbers for the resource.	
Email	Type the email address for the resource.	
Notes	Type any additional notes on the resource.	
Available Resources	Review resources that have already been defined.	
New	Add further resources.	
Save	Add a new resource to the Available Resource list.	
Delete	Delete a resource from the Available Resources list.	

Notes

- You can transport these resource definitions between models, using the Export Reference Data and Import Reference Data options

Learn more

- [Export Reference Data](#)^[306]
- [Import Reference Data](#)^[308]

5.10.2.4 Project Clients

Project clients are the eventual owners of the software system.

Access [Settings](#) | [Project Types](#) | [People](#) > [Project Clients](#)

Use to

- Capture client details associated with the current model

Reference

Field	Usage	See also
Name	Type or select the name of the client.	
Organization	Type the name of the organization that employs the client.	
Role(s)	Type the role the client plays in the project (for example, Manager, Sponsor).	
Phone 1, Phone 2, Mobile, Fax	Type the contact telephone numbers for the client.	
Email	Type the email address of the client.	
Notes	Type any additional notes on the client.	
Defined Clients	Review clients that have already been defined.	
New	Add details of further clients.	
Save	Add a new client to the Defined Client list.	
Delete	Delete a client record from the Defined Client list.	

Notes

- You can transport these client definitions between models, using the Export Reference Data and Import Reference Data options

Learn more

- [Export Reference Data](#) 

- [Import Reference Data](#)^[308]

5.10.3 General Types

The General Types dialog enables you to configure:

- Status Types
- Constraint Types
- Constraint Status Types
- Requirement Types
- Scenario Types

Access [Settings | Project Types | General Types](#)

Learn more

- [Status Types](#)^[995]
- [Constraint Types](#)^[997]
- [Constraint Status Types](#)^[998]
- [Requirement Types](#)^[998]
- [Scenario Types](#)^[999]

5.10.3.1 Status Types

You can configure a basic list of status types used in Enterprise Architect. Note that whilst most dialogs use this list, not all do so.

Access [Settings | Project Types | General Types > Status](#)

Use to

- Configure Status Types

Reference

Field	Usage	See also
Status	Indicate the name of the status.	
Description	Indicate a short description for the status.	
Status Type Color	Enable a color to be applied to a Status Type.	
Preview	Provide a visual representation of a Status.	

Restore Default	Restore the Status Type Color to it's default.	
Applies to...	By default, status colors only apply to Requirement, Issue and Change elements. You might decide to also apply these colors to other UML elements, such as Use Cases or Classes. To do this, click on the Applies to... button and select the checkbox against each required element type in the Applied Status Colors list.	Requirement ^[1471] , Issue ^[2244] Change ^[2245]
New	Create a new status.	
Save	Save the status, and add the status to the status list.	
Delete	Delete a status that has been highlighted.	
Type List	This list details the current status types and descriptions for each status.	

Notes

- You can transport the status types (and the colors assigned to status types) between models, using the Export Reference Data and Import Reference Data options
- To ensure status colors display on your diagrams, open the Options dialog at the Objects page and select the **Show status colors on diagrams** checkbox
- Requirement, Feature, Issue and Change elements have a status color compartment, but other elements do not; the status color for these elements is applied to the element shadow, therefore on the Options dialog Diagram Appearance page you must also select the **Element Shadows on** checkbox

Learn more

- [Export Reference Data](#)^[306]
- [Import Reference Data](#)^[308]
- [Object Appearance](#)^[556]
- [Diagram Appearance](#)^[550]

5.10.3.2 Constraint Types

The Constraint tab of the General Types dialog enables you to define constraints. These are picked up in a variety of places where constraints might fall into more categories than the basic (default) *Pre-*, *Post-* and *Invariant* conditions.

Access [Settings | Project Types | General Types > Constraint](#)

Use to

- Allows for definition of constraints

Reference

Field	Usage	See also
Constraint	Indicates the name of the constraint.	
Description	Indicates a brief description of the constraint.	
Note	Indicates any additional information required.	
Defined Constraint Type	A list of defined constraint types.	
New	Add a new constraint.	
Save	Saves the constraint and adds it to the Defined Constraint Type list.	
Delete	Removes a constraint from the Defined Constraint Type list.	

Notes

- You can transport these constraints between models, using the Export Reference Data and Import Reference Data options

Learn more

- [Export Reference Data](#) ^[306]
- [Import Reference Data](#) ^[308]

5.10.3.3 Constraint Status Types

You can configure the basic list of *constraint status types* used in Enterprise Architect.

Access [Settings | Project Types | General Types > Constraint Status](#)

Use to

- Allows for configuration of basic constraint status types


Reference

Field	Usage	See also
Status	Indicates the name of the constraint status.	
Status List	A list of defined constraint types.	
New	Add a new constraint status.	
Save	Saves the constraint status and adds it to the Status list.	
Delete	Removes a constraint status from the Status list.	

Notes

- You can transport these constraint status types between models, using the Export Reference Data and Import Reference Data options

Learn more

- [Export Reference Data](#) 
- [Import Reference Data](#) 

5.10.3.4 Requirement Types

The Requirement tab of the General Types dialog enables you to specify the generic set of requirement types that can be entered into the requirements sections of dialogs. This helps to maintain a single set of typed requirements.

Access [Settings | Project Types | General Types > Requirement](#)

Use to

- Specify the generic set of requirement types

Reference

Field	Usage	See also
Requirement	Indicates the name of the requirement.	
Description	Indicates a short description of the requirement.	
Weight	Indicates a weighting to apply to the requirement type.	
Defined Requirement Types	Displays all of the pre-defined and saved requirement types.	
New	Adds a new requirement.	
Save	Saves the requirement details and adds it to the Defined Requirement Types list.	
Delete	Deletes a requirement from the Defined Requirement Types list.	

Notes

- You can transport these requirement types between models, using the Export Reference Data and Import Reference Data options

Learn more

- [Export Reference Data](#) ³⁰⁸
- [Import Reference Data](#) ³⁰⁸

5.10.3.5 Scenario Types

A drop-down list of scenario types is available in the Scenario tab of an element Properties dialog, with the standard types *Basic Path*, *Exception* and *Alternate Flow*. You can set additional scenario types using the Scenario tab the General Types dialog.

Access [Settings | Project Types | General Types > Scenario](#)

Use to

- Add additional Scenario Types

Reference

Field	Usage	See also
Scenario Type	Indicates the name of the scenario.	
Description	Indicates a short description of the scenario.	
Weight	Indicates a weighting to apply to the scenario type.	
Note	Indicates any additional information on the scenario type.	
Defined Scenario Types	Displays all of the pre-defined and saved scenario types.	
New	Add a new scenario.	
Save	Saves the scenario details and adds it to the Defined Scenario Types list.	
Delete	Deletes a scenario from the Defined Scenario Types list.	

Notes

- You can transport these scenario types between models, using the Export Reference Data and Import Reference Data options

Learn more

- [Scenarios](#)^[852]
- [Export Reference Data](#)^[306]
- [Import Reference Data](#)^[308]

5.10.3.6 Metrics and Estimation

TCF values, EFC values and Default Hour Rate for a project are controlled from the Estimation Factors dialog.

Risk, metric and effort types for a project are controlled from the Project Indicators dialog.

For further information on these see the [Project Management](#)^[447] and [Resource Management](#)^[449] topics, or specifically:

- [Technical Complexity Factors](#)^[517]

- [Environment Complexity Factors](#) ^[518]
- [Default Hours](#) ^[520]
- [Effort Types](#) ^[456]
- [Metric Types](#) ^[457]
- [Risk Types](#) ^[458]

5.10.4 Maintenance

Access [Settings](#) | [Project Types](#) | [Maintenance](#)

Use to

- Set the base Problem Types that are handled

Learn more

- [Testing Types](#) ^[1002]

5.10.4.1 Problem Types

NOT CURRENTLY USED

For the maintenance and change control screens, you can use the Maintenance dialog to set the base Problem Types that are handled. Examples are hardware-related issues, performance problems, software bugs and network problems.

Access [Settings](#) | [Project Types](#) | [Maintenance](#) > [Problem Types](#)

Use to

- Set the base Problem Types that are handled

Reference

Field	Usage	See also
Problem Type	Indicates the name of the problem type.	
Description	Indicates a short description of the problem type.	
Weight	Indicates a weighting to apply to the problem type.	

Note	Indicates any additional information on the problem type.	
Defined Types	Displays all of the pre-defined and saved problem types.	
New	Add a new problem type.	
Save	Saves the scenario details and adds it to the Defined Types list.	
Delete	Deletes a scenario from the Defined Types list.	

Notes

- You can transport these problem types between models, using the Export Reference Data and Import Reference Data options
- You transport the problem types together with test types as a *Maintenance Types* file

Learn more

- [Export Reference Data](#) ^[306]
- [Import Reference Data](#) ^[308]

5.10.4.2 Testing Types

Use the Test Types tab of the Maintenance dialog to add testing types to the basic set that comes with Enterprise Architect. Typical test types are load tests, performance tests and function tests.

Access [Settings | Project Types | Maintenance > Test Types](#)

Use to

- Add Testing Types

Reference

Field	Usage	See also
Test Type	Type the name of the test type.	
Description	Type a short description of the test type.	
Weight	Type a weighting to apply to the test type.	

Note	Type any additional information on the test type.	
Defined Types	Displays all of the pre-defined and saved test types.	
New	Add a new test type.	
Save	Saves the scenario details and adds it to the Defined Types list.	
Delete	Deletes a scenario from the Defined Types list.	

Notes

- You can transport these test types between models, using the Export Reference Data and Import Reference Data options
- You can either export the test types together with the default problem types, as a *Maintenance Types* file, or separately as a *Test Types* file

Learn more

- [Export Reference Data](#)^[306]
- [Import Reference Data](#)^[308]

5.10.5 Resources

The Resources window displays a tree of MDG Technologies, Templates, Documents, UML Profiles and Patterns, commonly-used model elements and Relationship Matrix profiles. This view provides useful shortcuts and re-use functions that you can use to add stock elements to the current model, and patterns and elements for additional information.

Access [View](#) | [More Project Tools](#) | [Project Resources](#)

Use to

- Add Stock elements to the current model
- Add patterns and elements

Reference

Field	Usage	See also
MDG Technologies	Provides a convenient way to insert complex new elements and features without having to retype or reconfigure each element.	MDG Technologies ^[1283]
Templates	Provides a range of templates for creating HTML (web) reports, RTF reports in either the legacy report generator or the extended RTF report generator, linked documents and MDG Technology reports. You can create, edit, copy and delete your own templates, and view and copy system-supplied or technology-supplied templates using context menu options.	Web Style Templates ^[2355] RTF Documents ^[2257] The Legacy RTF Report Generator ^[2318] Generate RTF Documents ^[2253] Linked Documents ^[928]
Documents	Provides shortcuts to saved RTF documents.	
Matrix Profiles	Provides quick access to saved Relationship Matrix profiles; double-click on a profile to load the matrix with the saved settings and source-target packages.	Matrix Profiles ^[643]
Favorites	Provides shortcuts to selected elements that you want to access quickly.	
Stylesheets	Enables you to import XSL Style sheets, which are then available in the drop-down list on the XML Export dialog.	Publish Model Package ^[414]
UML Profiles	Provides a convenient way to insert complex new elements and features without having to retype or reconfigure each element.	UML Profiles ^[1277]
UML Patterns	Provides a convenient way to insert complex new elements and features without having to retype or reconfigure each element.	Design Patterns ^[1277]

Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Configure Resources** permission to maintain Resources window items
- From release 7.5 of Enterprise Architect, a method of importing MDG Technologies into the Resources window is available but *Not Recommended*; however, you might previously have imported Technologies into the Resources window, and these are still available until you specifically delete them (right-click on the Technology and select the **Delete Technology** context menu option) - the

preferred method of making MDG Technologies available to your users is to import them into the % APPDATA% folder

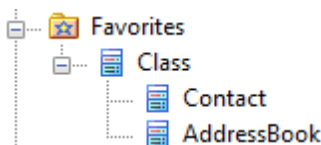
- You can also synchronize the Tagged Values and constraints for any elements created from a profile element in the Resources window
- To add a document to the shortcut list, select the **Project | Documentation | Rich Text Format (RTF) Report** menu option; once you have defined your document click on the **Resource Document** button and type in a name - the document name then displays in the Resources window; by right-clicking on the document name you can regenerate documents individually or as a batch, or open them directly from Enterprise Architect
- If you select a style sheet on export, Enterprise Architect applies that style sheet to the XML generated before saving to file; this makes it convenient to generate other forms of output from the base XML content - combined with UML Profiles, this is a powerful means of extending Enterprise Architect to generate almost any content required

Learn more

- [Permission List](#)^[264]
- [importing MDG Technologies](#)^[1288]
- [Synchronize Tagged Values and Constraints](#)^[1281]
- [Resource Documents](#)^[2259]

5.10.5.1 Favorites

The Resources window contains a *Favorites* folder. Here you can hyperlink to any UML element from the model as a whole, and conveniently drag and drop instances or links to this element into other diagrams. This is particularly useful where certain elements - such as the list of Actors in a system - are re-used again and again, and switching to the *Actors* folder is not convenient. In cases like this, using the *Favorites* folder makes managing and creating your model much easier.



Topics

Topic	Detail	See also
Add to the Favorites Folder	<p>To add an element to the Favorites folder:</p> <ul style="list-style-type: none"> • In a diagram, right-click on the element to add • From the context menu select the Find Add to Favorites option • Switch back to the Resources window and check the <i>Favorites</i> folder; the new element should be listed in its category within the favorites 	

Topic	Detail	See also
Delete from the Favorites Folder	To delete a favorite: <ul style="list-style-type: none">• Right-click on it within the <i>Favorites</i> folder in the Resources window• Select Delete Favorite from the context menu• Confirm the action by clicking on the Yes button	
View Properties of a Favorite	To view a favorite's properties from the <i>Favorites</i> folder: <ul style="list-style-type: none">• Select and right-click on the favorite in the Resources window• Select Element Properties from the context menu	

Learn more

- [Resources](#) ¹⁰⁰³

Part

VI

6 Standard UML Models



This topic provides an introduction to Enterprise Architect's diagrams, elements and connectors, and its modeling process.

It also illustrates its alignment to the Unified Modeling Language (UML) 2.3, an open modeling standard, defined and maintained by the Object Management Group.

Topic	Detail	See also
The Unified Modeling Language (UML)	<p>The UML standard defines notations and rules for specifying business and software systems; the notation supplies a rich set of graphic elements for modeling object oriented systems, and the rules state how those elements can be connected and used.</p> <p>UML is not a tool for creating software systems; instead, it is a visual language for communicating, modeling, specifying and defining systems.</p> <p>UML is not a prescriptive process for modeling software systems; it does not supply a method or process, simply the language. You can therefore use UML in a variety of ways to specify and develop your software engineering project.</p> <p>This language is designed to be flexible, extendable and comprehensive, yet generic enough to serve as a foundation for all system modeling requirements. With its specification, there is a wide range of elements characterized by the kinds of diagrams they serve, and the attributes they provide. All can be further specified by using stereotypes, Tagged Values and profiles.</p> <p>Enterprise Architect supports many different kinds of UML elements (as well as some custom extensions); together with the connectors between elements, these form the basis of the model.</p>	<p>UML Diagrams ^{f1010}</p> <p>UML Elements ^{f1090}</p> <p>UML Connectors ^{f1206}</p> <p>UML Stereotypes ^{f1264}</p> <p>UML Profiles ^{f1277}</p> <p>Tagged Values ^{f969}</p>
Wide Range of Applications	<p>Although initially conceived as a language for software development, UML can be used to model a wide range of real world domains and processes (in business, science, industry, education and elsewhere), organizational hierarchies, deployment maps and much more.</p> <p>Enterprise Architect also provides additional custom diagrams and elements, to address further modeling interests.</p>	<p>Modeling Basics ^{f657}</p>
Extending UML for New Domains	<p>Using UML Profiles, UML Patterns, Grammars, Data Types, Constraints, MDG Technologies and other extensions, UML</p>	<p>UML Profiles ^{f1277}</p>

Topic	Detail	See also
	<p>and Enterprise Architect can be tailored to address a particular modeling domain not explicitly covered in the original UML specification</p> <p>Enterprise Architect makes extending UML simple and straightforward and, best of all, the extension mechanism is still part of the UML Specification.</p>	<p>Design Patterns ^[1271]</p> <p>MDG Technologies ^[1283]</p>
Recommended Reading	<p>In addition to the UML Specification available from the OMG, two books that provide excellent introductions to UML are:</p> <ul style="list-style-type: none"> • <i>Schaum's Outlines: UML</i> by Bennet, Skelton and Lunn Published by McGraw Hill. ISBN 0-07-709673-8 • <i>Developing Software with UML</i> by Bern Oestereich Published by Addison Wesley. ISBN 0-201-36826-5 	

Learn more

- [OMG website](#)
- Diagram [Toolbox](#) ^[698] descriptions
- The *EAExample.eap* project supplied with Enterprise Architect
- Online [UML Tutorial](#) (parts 1 and 2) and [UML 2.0 Tutorial](#)

6.1 UML Diagrams

A UML diagram is a representation of the components or elements of a system or process model and, depending on the type of diagram, how those elements are connected or how they interact from a particular perspective. For example, how and why an object changes state, or how requirements are realized by the process or a system.

Topics

Topic	Detail	See also
Working with Diagrams	<p>Diagrams are developed in the main workspace, in which you create and connect model elements.</p> <p>You create diagrams by right-clicking a package and selecting the New Diagram context menu option, or load them by double-clicking on their diagram icon in the Project Browser.</p>	Diagram Tasks ^[72]
Types of Diagram	<p>There are two major groupings of UML diagrams:</p> <ul style="list-style-type: none"> • Structural Diagrams which depict the structural elements composing a system or function, reflecting the static relationships of a structure, or run-time architectures • Behavioral Diagrams which show a dynamic view of the model, depicting the behavioral features of a system or business process 	Structural Diagrams ^[101] Behavioral Diagrams ^[102]
Extended Diagrams	Enterprise Architect provides a set of additional diagram types that extend the core UML diagrams for domain-specific models.	Extended Diagrams ^[150]
Custom Diagrams	Enterprise Architect also supports diagram types specific to MDG Technologies, including integrated technologies.	Specialized UML Models ^[149]

6.2 UML Structural Models

Structural diagrams depict the structural elements composing a system or function. These diagrams reflect the static relationships of a structure, such as Class or Package diagrams, or run-time architectures such as Object or Composite Structure diagrams.

Structural diagrams include the following diagram types:

Diagram Type	Detail	See also
Class Diagrams	Class diagrams capture the logical structure of the system, the Classes and objects that make up the model, describing what exists and what attributes and behavior it has.	Class diagrams ^[1014]
Composite Structure Diagrams	Composite Structure diagrams reflect the internal collaboration of Classes, Interfaces and Components (and their properties) to describe a functionality.	Composite Structure diagrams ^[1017]
Component Diagrams	Component diagrams illustrate the pieces of software, embedded controllers and such that make up a system, and their organization and dependencies.	Component diagrams ^[1023]
Deployment Diagrams	Deployment diagrams show how and where the system is to be deployed; that is, its execution architecture.	Deployment diagrams ^[1020]
Object Diagrams	Object diagrams depict object instances of Classes and their relationships at a point in time.	Object diagrams ^[1015]
Package Diagrams	Package diagrams depict the organization of model elements into packages and the dependencies amongst them.	Package diagrams ^[1011]
Profile Diagrams	Profile Diagrams are those created in a «profile» package, to extend UML elements, connectors and components.	Profile Diagrams ^[1025]

6.2.1 Package Diagram

Package diagrams depict the organization of model elements into packages and the dependencies amongst them, including package imports and package extensions. They also provide a visualization of the corresponding namespaces.




















Use to

- Depict the organization of model elements into packages
- Show dependencies between packages
- Visualize namespaces

[Example Diagram](#) [Example Package Diagram](#) ^[1013]

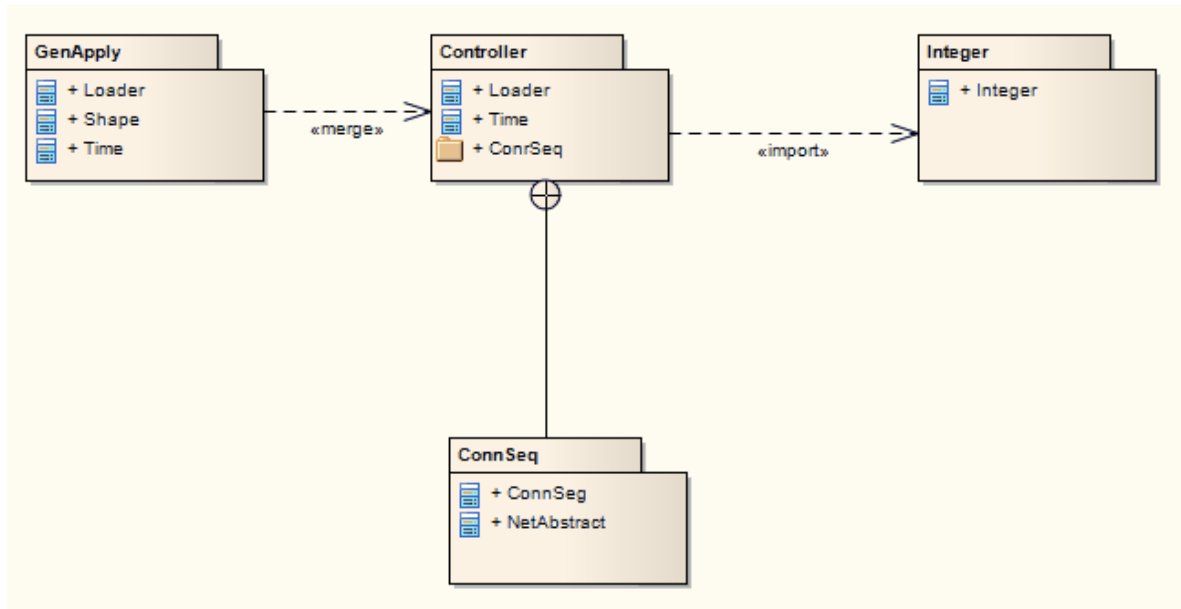
Tools

Select Package diagram elements and connectors from the Class pages of the Toolbox.

Package Diagram Elements	Package Diagram Connectors
 Package	 Associate
 Class	 Generalize
 Interface	 Compose
 Data Type	 Aggregate
 Enumeration	 Association Class
 Primitive	 Assembly
 Table	 Realize
 Signal	 Nesting
 Association	 Package Merge
	 Package Import

6.2.1.1 Example Package Diagram

The following example demonstrates a basic Package diagram.



Item	Description	See also
Nesting connector between <i>ConnSeq</i> and <i>Controller</i>	The nesting connector between <i>ConnSeq</i> and <i>Controller</i> reflects what the package contents reveal. Package contents can be listed by clicking on the diagram background to display the diagram's Properties dialog, selecting the Elements tab and selecting the Package Contents checkbox.	Properties ⁷²⁵ dialog
«import» connector	The «import» connector indicates that the elements within the target <i>Integer</i> package, which in this example is the single Class <i>Integer</i> , are imported into the package <i>Controller</i> . The <i>Controller's</i> namespace gains access to the <i>Integer</i> Class; the <i>Integer</i> namespace is not affected.	
«merge» connector	The «merge» connector indicates that the package <i>Controller's</i> elements are imported into <i>GenApply</i> , including <i>Controller's</i> nested and imported contents. If an element already exists within <i>GenApply</i> , such as <i>Loader</i> and <i>Time</i> , these elements' definitions are expanded by those included in the package <i>Controller</i> . All elements added or updated by the merge are noted by a generalization relationship back to that package.	

Item	Description	See also

6.2.2 Class Diagram

The Class diagram captures the logical structure of the system - the Classes - and things that make up the model. It is a static model, describing what exists and what attributes and behavior it has, rather than how something is done.




















Use to

- Illustrate relationships between Classes and Interfaces; Generalizations, Aggregations and Associations are all valuable in reflecting inheritance, composition or usage, and connections, respectively

Example Diagram [Example Class Diagram](#) ^[1015]

Tools

Select Class diagram elements and connectors from the Class pages of the Toolbox.

Class Diagram Elements	Class Diagram Connectors
 Package	 Associate
 Class	 Generalize
 Interface	 Compose
 Data Type	 Aggregate
 Enumeration	 Association Class
 Primitive	 Assembly
 Table	 Realize
 Signal	 Nesting
 Association	 Package Merge
	 Package Import

Learn more

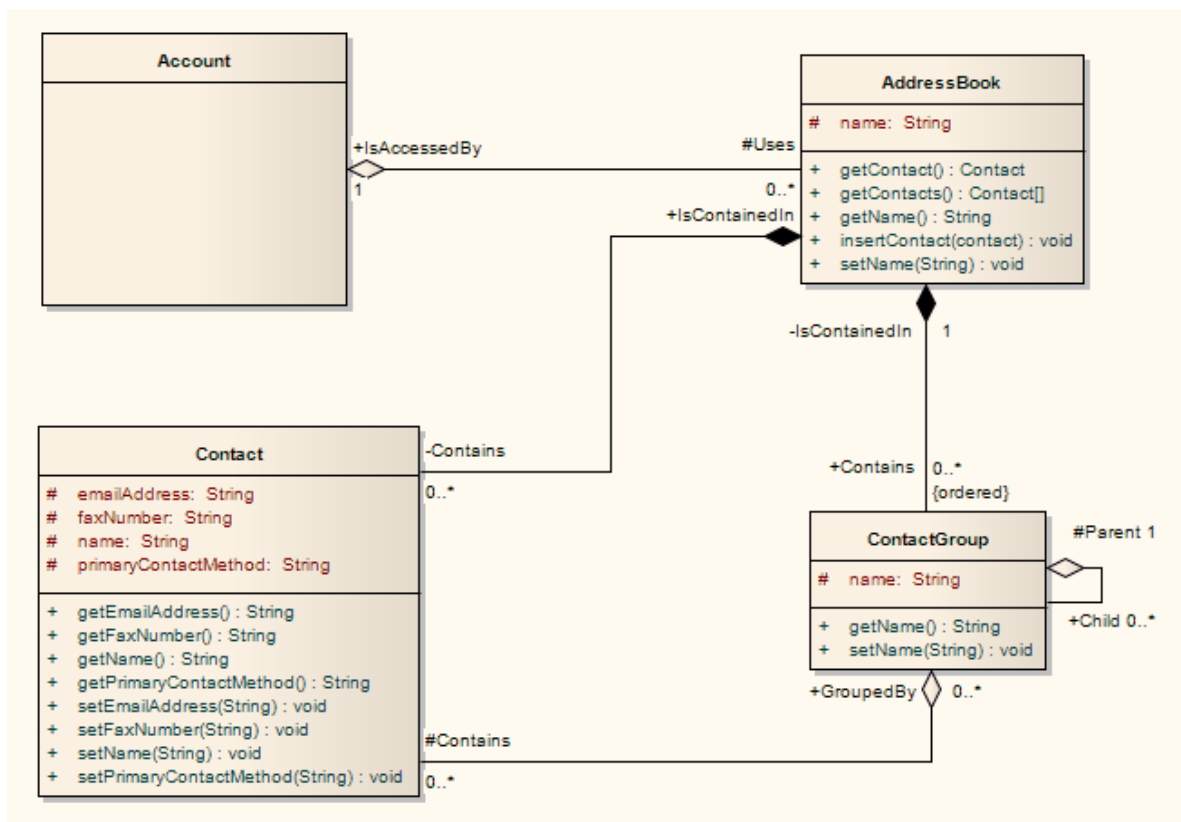
- [Active Classes](#) ^[1182]
- [Parameterized \(template\) Classes](#) ^[1183]

- [Stereotyped Business Interaction Elements](#) ^[1514]

6.2.2.1 Example Class Diagram

There are two forms of the [Aggregation](#) ^[1208] relationship in the following diagram:

- The pale form indicates that the Class *Account* uses *AddressBook*, but does not necessarily contain *AddressBook*
- The dark *Composite* Aggregation form indicates ownership or containment by the target Classes (at the diamond end) of the source Classes (at the arrow end)



6.2.3 Object Diagram

An Object diagram is closely related to a Class diagram, with the distinction that it depicts object instances of Classes and their relationships at a point in time. This might appear similar to a Composite Structure diagram, which also models run-time behavior; the difference is that Object diagrams exemplify the static Class diagrams, whereas Composite Structure diagrams reflect run-time architectures different from their static counterparts.

Object diagrams do not reveal architectures varying from their corresponding Class diagrams, but reflect multiplicity and the roles instantiated Classes could serve.

Use to












- Object diagrams are useful in understanding a complex Class diagram, by creating different cases in which the relationships and Classes are applied

- An Object diagram can also be a kind of Communication diagram (which also models the connections between objects, but additionally sequences events along each path)

Example Diagram [Example Object Diagram](#) [1016]

Tools

Select Object diagram elements and connectors from the Object pages of the Toolbox.

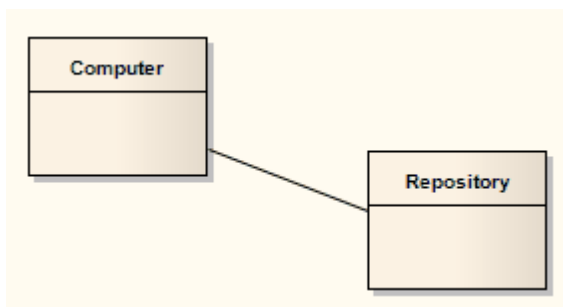
Object Diagram Elements	Object Diagram Connectors
 Actor	 Information Flow
 Object	 Associate
 Collaboration	 Dependency
 Collaboration Use	
 Information Item	
 Boundary	
 Control	
 Entity	

Learn more

- [Business Modeling Objects](#) [1514]
- [Communication diagram](#) [1084]

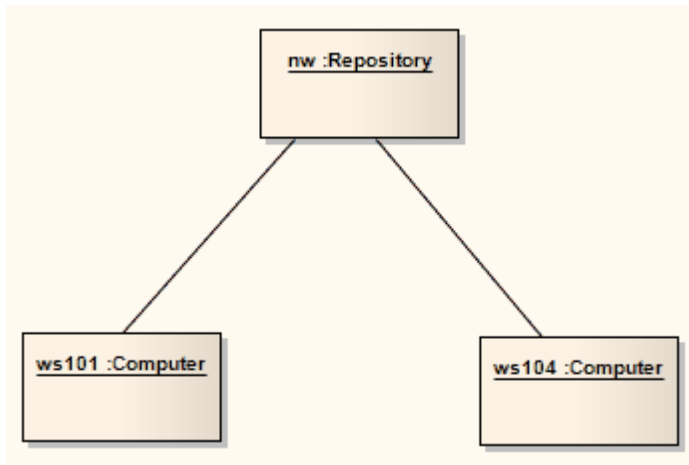
6.2.3.1 Example Object Diagram

The following example first shows a simple Class diagram, with two Class elements connected.



The Classes above are instantiated below as Objects in an Object diagram. There are two instances of *Computer* in this model, which can prove useful for considering the relationships and interactions Classes

play in practice, as Objects.



6.2.4 Composite Structure Diagram

A Composite Structure diagram reflects the internal collaboration of Classes, Interfaces or Components (and their Properties) to describe a functionality. Composite Structure diagrams are similar to Class diagrams, except that they model a specific usage of the structure. Class diagrams model a static view of Class structures, including their attributes and behaviors.

In a Composite Structure diagram, Classes are accessed as Parts or run-time instances fulfilling a particular role. These Parts can have multiplicity, if the role filled by the Class requires multiple instances. Ports defined by a Part's Class should be represented in the composite structure, maintaining that all connecting Parts provide the required interfaces specified by the Port. There is extensive flexibility, and an ensuing complexity, that come with modeling composite structures. To optimize your modeling, consider building Collaborations to represent reusable patterns responding to your design issues.









Use to






- Express run-time architectures, usage patterns and the participating elements' relationships, which might not be reflected by static diagrams.

Example Diagram [Example Composite Structure Diagram](#) 

Tools

Select Composite Structure diagram elements and connectors from the Composite pages of the Toolbox.

Composite Structure Diagram Elements	Composite Structure Diagram Connectors
 Class	 Connector
 Interface	 Assembly
 Part	 Role Binding
 Port	 Represents

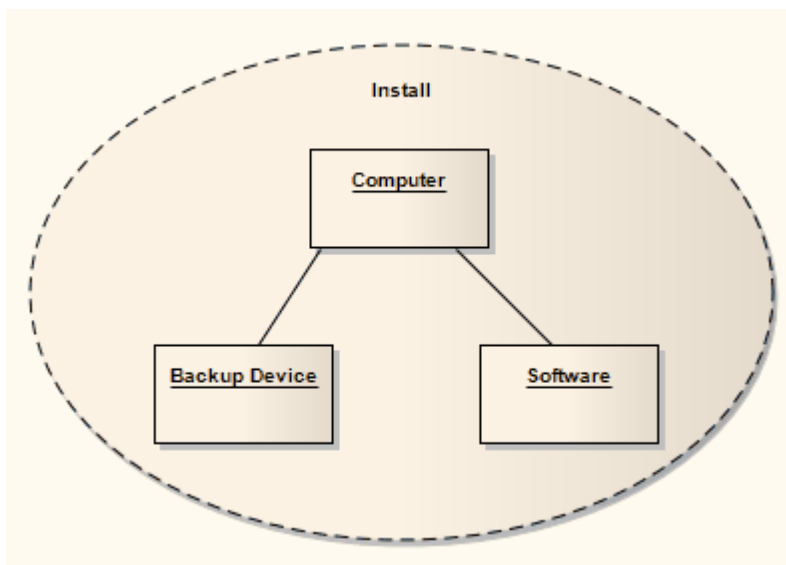
Composite Structure Diagram Elements	Composite Structure Diagram Connectors
 Collaboration	 Occurrence
 Collaboration Use	 Delegate
 Expose Interface	

Learn more

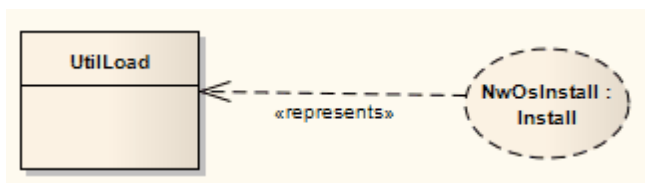
- [Business Use Case Realization](#) ^[1514]

6.2.4.1 Example Composite Structure Diagram

The following diagram shows a Collaboration used in Composite Structure diagrams to model common patterns. This particular example shows a relationship for performing an installation.



The following diagram uses the *Install* Collaboration in a Collaboration Use, and applies it to the *UtilLoad* Class via a «represents» relationship. This indicates that the classifier *UtilLoad* uses the collaboration pattern within its implementation.



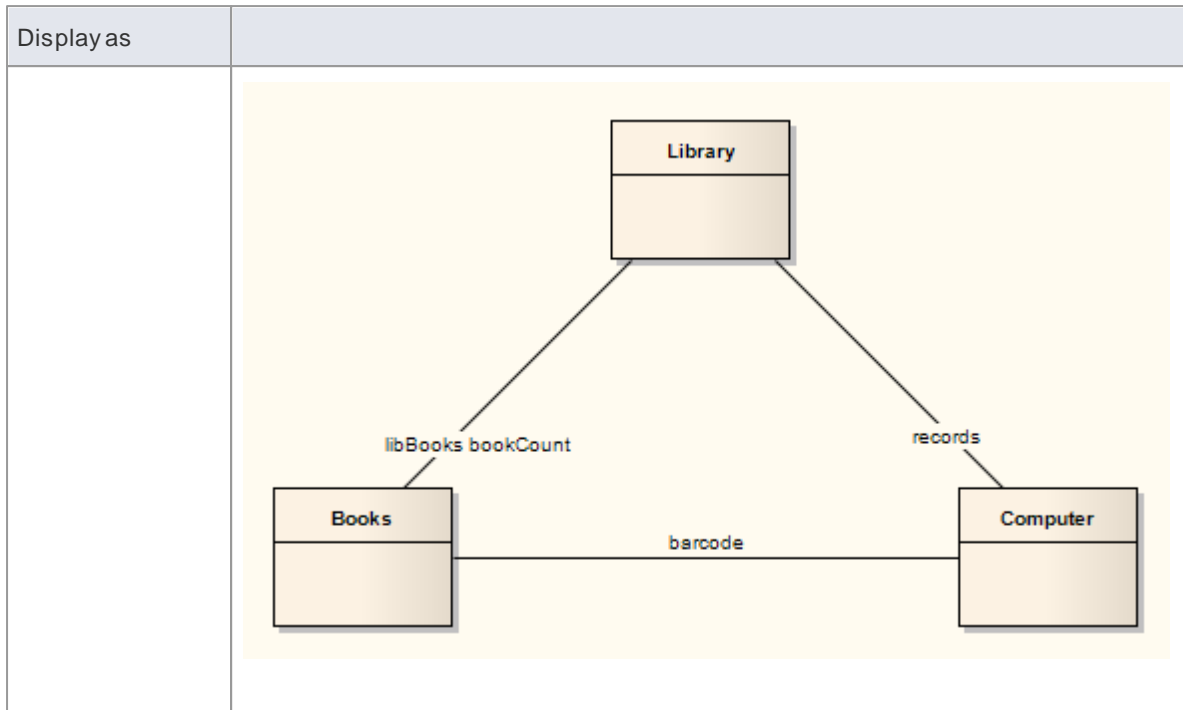
Learn more

- [Collaboration Use](#) ^[1186]

6.2.4.2 Properties

A *property* is a nested structure within a classifier, which is usually a Class or an Interface, on a Composite Structure diagram. The contained structure reflects instances and relationships reflected within the containing classifier. Properties can have multiplicity.

Display as	
Parts	<div data-bbox="486 521 1433 1014" style="border: 1px solid black; padding: 10px; background-color: #f9f9f9;"> <pre> classDiagram class Library class Books["libBooks :Books [bookCount]"] class Computer["records :Computer"] Library --> Books Library --> Computer Books -- Computer : barcode </pre> </div> <p>There are two Parts, <i>libBooks</i> and <i>records</i>, which are instances corresponding to the Classes <i>Books</i> and <i>Computer</i> respectively.</p> <p>After dragging Parts from the Toolbox out to the workspace, right-click on a Part and select the Advanced Set Property Type context menu option to connect to a classifier.</p> <p>If Parts disappear when dragged onto the Class, adjust the Z-order of the Class (right-click on it and select the Z-Order context menu option).</p> <p>The relationship between the two Parts is indicated by the connector, reflecting that communication between the Parts is via the <i>barcode</i>. This contained structure and its Parts are properties owned by the <i>Library</i> Class.</p> <p>To indicate a property that is <i>not</i> owned by composition to the containing classifier, use a box symbol with a dashed outline, indicating association; to do this:</p> <ol style="list-style-type: none"> 1. Right-click on the Part and select the Properties context menu option. 2. Select the Advanced page of the Properties dialog. 3. Set the IsReference option to true.
Association Roles	<p>Properties can also be reflected using a normal composite structure (without containing it in a Class), with the appropriate connectors, parts and relationships indicated through connections to the Class.</p> <p>This alternative representation is shown in the following diagram; however, this depiction fails to express the ownership immediately reflected by containing properties within a classifier.</p>



6.2.5 Deployment Diagram











A Deployment diagram shows how and where the system is to be deployed; that is, its execution architecture.








Hardware devices, processors and software execution environments (system Artifacts) are reflected as Nodes, and the internal construction can be depicted by embedding or nesting Nodes. Deployment relationships indicate the deployment of Artifacts, and Manifest relationships reveal the physical implementation of components. As Artifacts are allocated to Nodes to model the system's deployment, the allocation is guided by the use of deployment specifications.

Example Diagram [Example Deployment Diagram](#) 

Tools

Select Deployment diagram elements and connectors from the Deployment pages of the Toolbox.

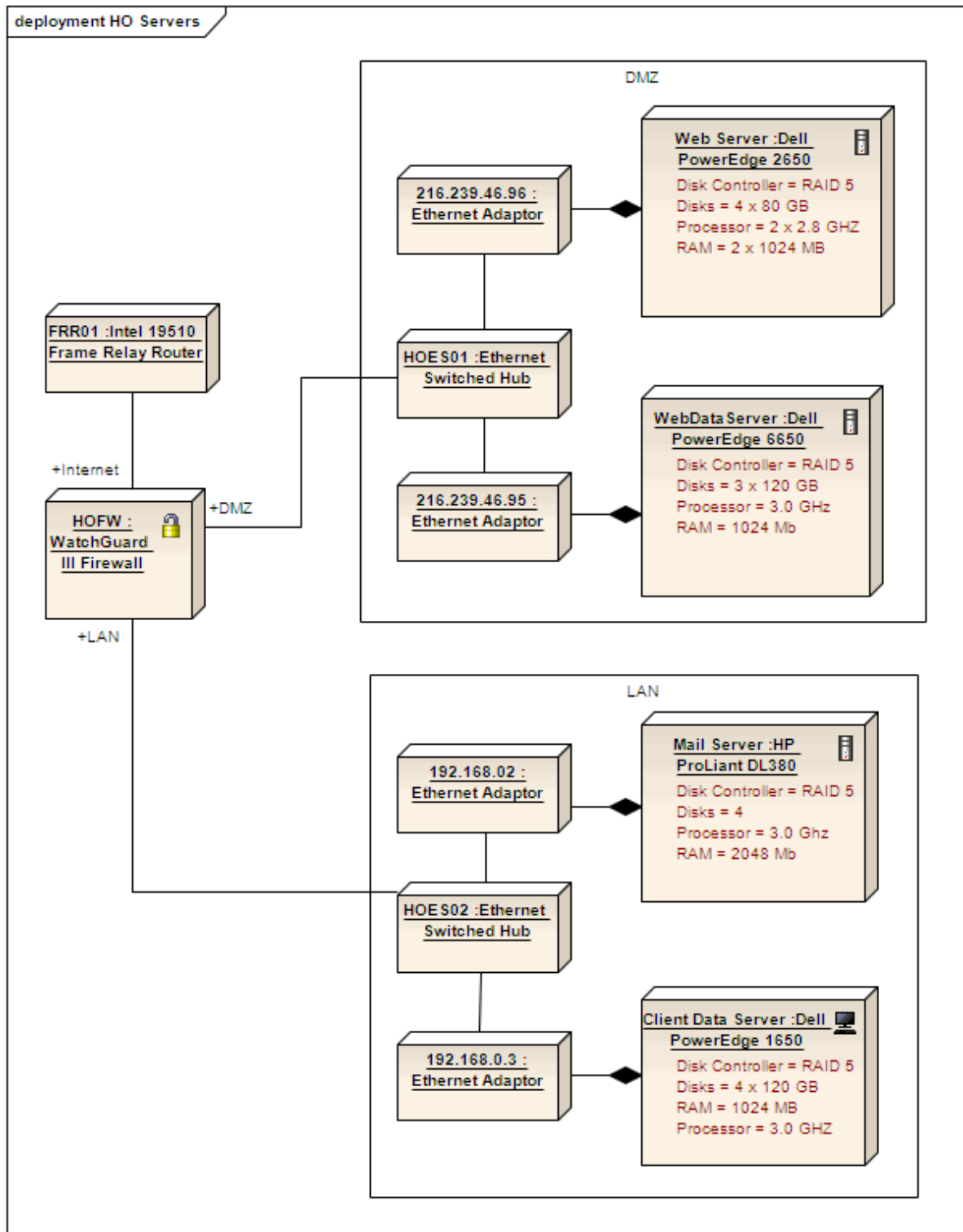
Deployment Diagram Elements	Deployment Diagram Connectors
 Node	 Associate
 Device	 Communication Path
 Execution Environment	 Association Class
 Component	 Generalize
 Interface	 Realize

Deployment Diagram Elements	Deployment Diagram Connectors
 Artifact	 Deployment
 Document Artifact	 Manifest
 Deployment Specification	 Nesting
 Package	

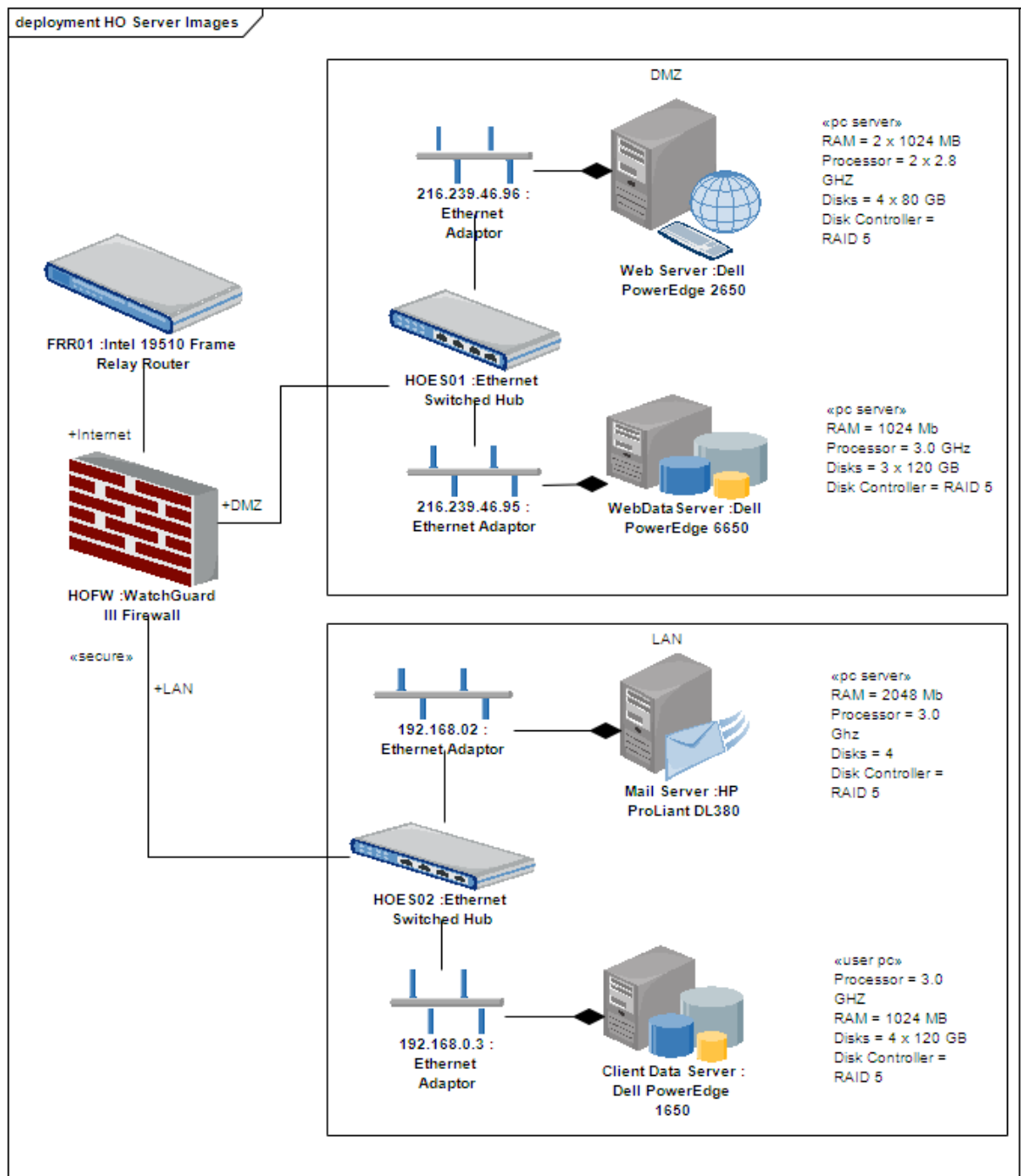
6.2.5.1 Example Deployment Diagram

A simple Deployment diagram is shown below, representing the arrangement of servers at a head office.

The servers are represented by Nodes linked by either simple or aggregate Association relationships.



Deployment diagrams are ideal for using alternative images for the objects that the elements represent. Such images can be substituted for the elements in the above diagram, as shown below:



Learn more

- [Alternative Images](#) ^[755]

6.2.6 Component Diagram
















A Component diagram illustrates the pieces of software, embedded controllers and such that make up a system, and their organization and dependencies.

A Component diagram has a higher level of abstraction than a Class diagram; usually a component is implemented by one or more Classes (or Objects) at runtime. They are building blocks, built up so that eventually a component can encompass a large portion of a system.

[Example Diagram](#) [Example Component Diagram](#) ^[1024]

Tools

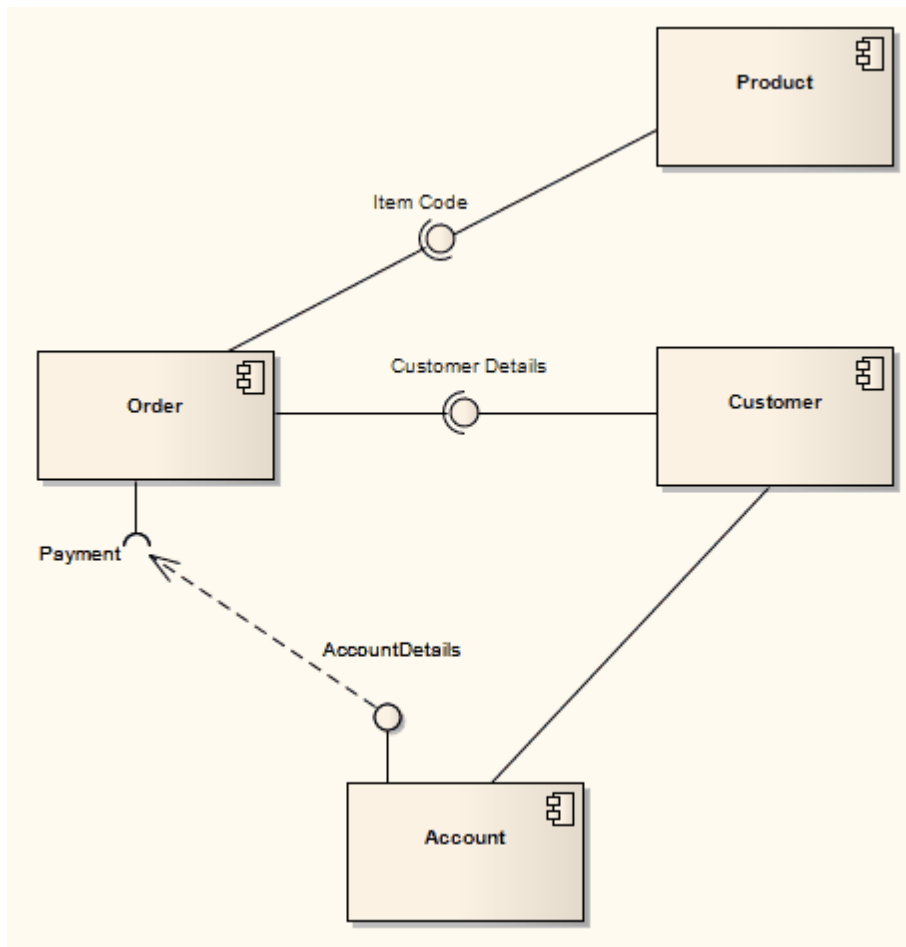
Select Component diagram elements and connectors from the Component pages of the Toolbox.

Component Diagram Elements	Component Diagram Connectors
 Package	 Assembly
 Packaging Component	 Delegate
 Component	 Associate
 Class	 Realize
 Interface	 Generalize
 Object	
 Port	
 Expose Interface	
 Artifact	
 Document Artifact	

6.2.6.1 Example Component Diagram

The following diagram demonstrates some components and their inter-relationships.

Assembly connectors connect the provided interfaces supplied by *Product* and *Customer* to the required interfaces specified by *Order*. A Dependency relationship maps a customer's associated *account details* to the required interface *Payment*, indicated by *Order*.



Learn more

- [Assembly](#)^[1209]
- [Dependency](#)^[1221]

6.2.7 Profile Diagram










A Profile diagram is any diagram created in a «profile» package.

Profiles provide a means of extending the UML. They are based on additional stereotypes and Tagged Values that are applied to UML elements, connectors and their components. A Profile is a collection of such extensions that together describe some particular modeling problem and facilitate modeling constructs in that domain.

Example Diagram [Example Profile Diagram](#)^[1026]

Tools

Select the following Profile diagram elements and connectors from the Profile pages of the Toolbox.

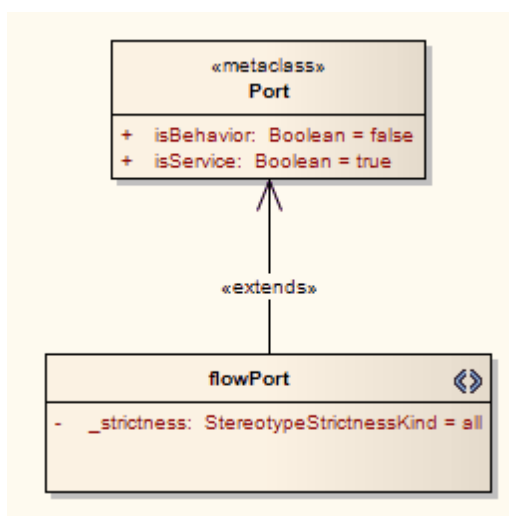
Profile Diagram Elements	Profile Diagram Connectors
 Profile	 Extension
 Stereotype	 Generalize
 Metaclass	 Application
 Enumeration	 Tagged Value
	 Redefinition

Learn more

- [Developing Profiles](#)^[1293]
- [Create Profiles](#)^[1295]

6.2.7.1 Example Profile Diagram

A typical unit on a Profile diagram resembles the following illustration:



6.3 UML Behavioral Models

Behavioral diagrams depict the behavioral features of a system or business process. Behavioral diagrams include the following diagram types:

Diagram Type	Detail	See also
Activity Diagrams	Activity diagrams model the behaviors of a system, and the way in which these behaviors are related in an overall flow of the system.	Activity diagrams ^[1028]
Use Case Diagrams	Use Case diagrams capture Use Cases and relationships among Actors and the system; they describes the functional requirements of the system, the manner in which external operators interact at the system boundary, and the response of the system.	Use Case diagrams ^[1030]
State Machine Diagrams	State Machine diagrams illustrate how an element can move between states, classifying its behavior according to transition triggers and constraining guards.	State Machine diagrams ^[1032]
Timing Diagrams	Timing diagrams define the behavior of different objects within a time-scale, providing a visual representation of objects changing state and interacting over time.	Timing diagrams ^[1051]
Sequence Diagrams	Sequence diagrams are structured representations of behavior as a series of sequential steps over time. They are used to depict work flow, message passing and how elements in general cooperate over time to achieve a result.	Sequence diagrams ^[1074]
Communication Diagrams	Communication diagrams show the interactions between elements at run-time, visualizing inter-object relationships.	Communication diagrams ^[1084]
Interaction Overview Diagrams	Interaction Overview diagrams visualize the cooperation between other interaction diagrams (Timing, Sequence, Communication and Interaction Overview diagrams) to illustrate a control flow serving an encompassing purpose.	Interaction Overview diagrams ^[1087]

Learn more

- [Behavioral Modeling](#) ^[899]
- [Code Generation from Behavioral Models](#) ^[1778]

6.3.1 Activity Diagram

Activity diagrams are used to model the behaviors of a system, and the way in which these behaviors are related in an overall flow of the system. The logical paths a process follows, based on various conditions, concurrent processing, data access, interruptions and other logical path distinctions, are all used to construct a process, system or procedure.

Use to


















- Model system behavior
- Model dynamic element interactions






Example Diagram [Example Activity Diagram](#)^[1029]

Tools

Select Activity diagram elements and connectors from the Activity pages of the Toolbox.

Click on the following elements and connectors for more information.

Activity Diagram Elements	Activity Diagram Connectors
 Activity	 Control Flow
 Structured Activity	 Object Flow
 Action	 Interrupt Flow
 Partition	
 Object	
 Central Buffer Node	
 Datastore	
 Decision	
 Merge	
 Send	
 Receive	
 Synch	
 Initial	
 Final	

Activity Diagram Elements	Activity Diagram Connectors
 Flow Final	
 Region	
 Exception	
 Fork/Join	
 Fork/Join	

Notes

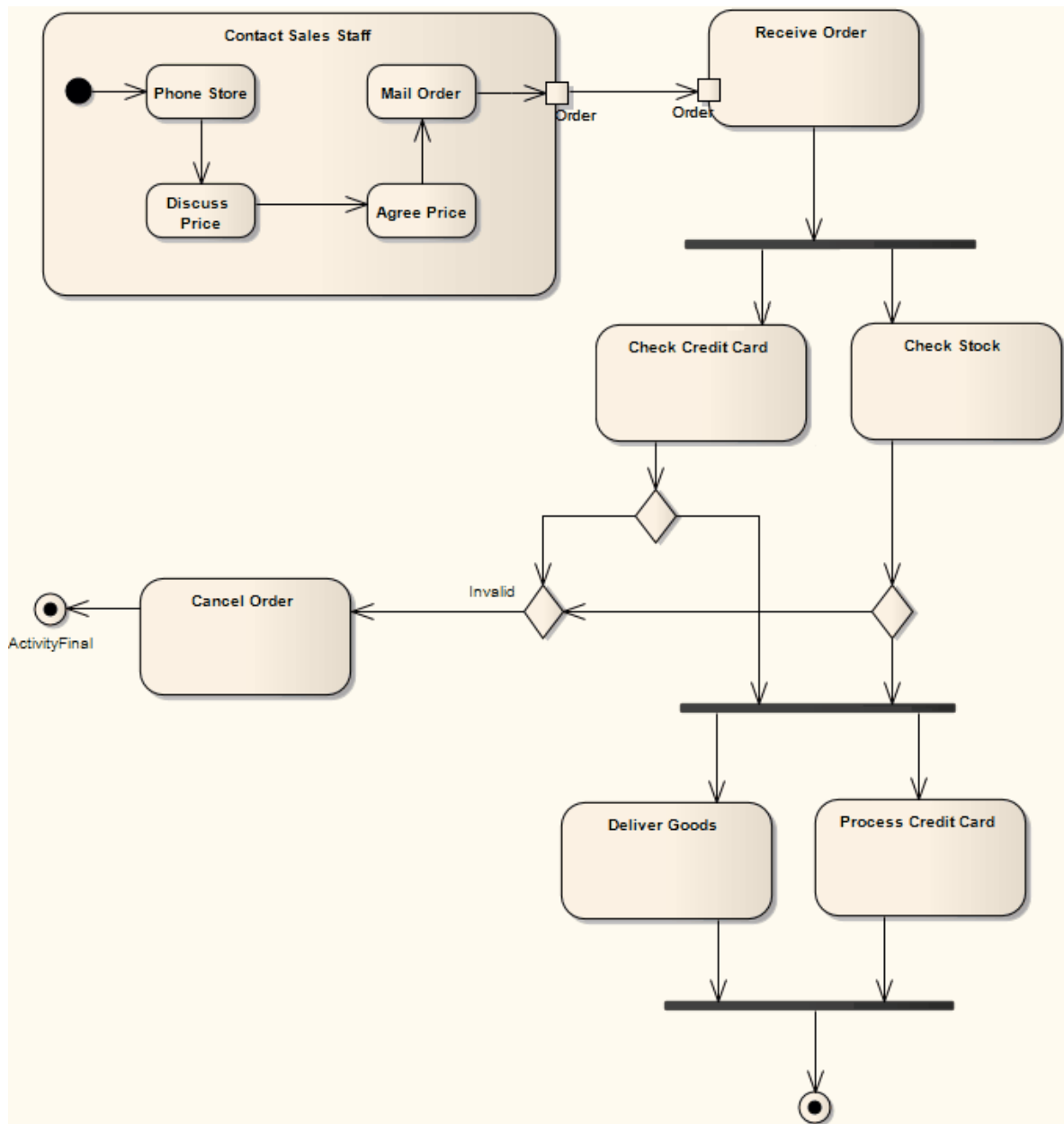
- You can create Analysis diagrams (Simplified Activity diagrams) containing the elements most useful for business process modeling, using the New Diagram dialog

Learn more

- [Analysis diagrams](#) ^[1509]
- [New Diagram](#) ^[724]
- [Activity pages](#) ^[712]

6.3.1.1 Example Activity Diagram

The following diagram illustrates some of the features of Activity diagrams, including Activities, Actions, Start Nodes, End Nodes and Decision points.



6.3.2 Use Case Diagram

Use Case diagrams capture Use Cases and the relationships between Actors and the subject (system).

Use to
















- Describe the functional requirements of the system
- Describe the manner in which outside things (Actors) interact at the system boundary
- Describe the response of the system.

Example Diagram [Example Use Case Diagram](#) ^[1032]

Tools

Select Use Case diagram elements and connectors from the Use Case pages of the Toolbox.

Click on the following elements and connectors for more information.

Use Case Diagram Elements	Use Case Diagram Connectors
 Actor	 Use
 Use Case	 Associate
 Test Case	 Generalize
 Collaboration	 Include
 Collaboration Use	 Extend
 Boundary	 Realize
 Package	 Invokes
	 Precedes

Notes

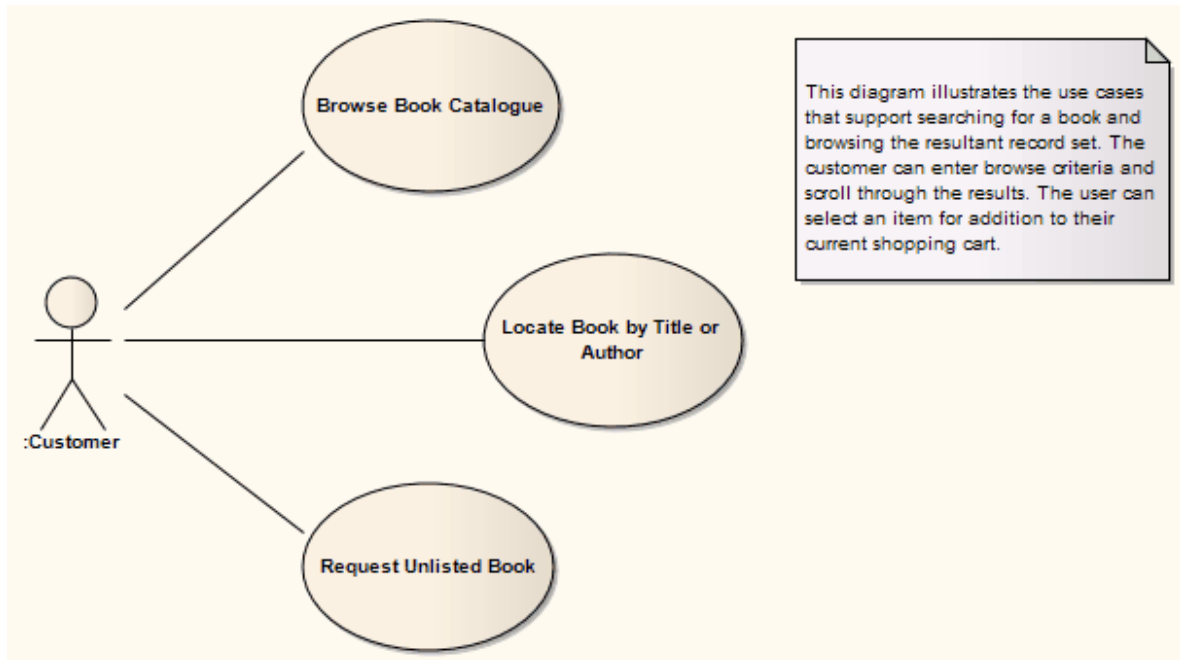
- Invokes and Precedes relationships are defined by the Open Modeling Language (OML); they are stereotyped Dependency relationships
- Invokes indicates that Use Case A, at some point, causes Use Case B to happen
- Precedes indicates that Use Case C must complete before Use Case D can begin

Learn more

- [Use Case Extension Points](#) ^[1176]
- [Use Rectangle Notation](#) ^[1177]
- [Business Use Case](#) ^[1514]
- [Use Case pages](#) ^[707]

6.3.2.1 Example Use Case Diagram

The following diagram illustrates some features of Use Case diagrams:



6.3.3 State Machines

State Machines illustrate how an element (often a Class) can move between states, classifying its behavior according to transition triggers and constraining guards.

Naming:

- State Machines were formerly known as State diagrams
- State Machine representations in UML are based on the Harel State Chart Notation and therefore are sometimes referred to as State Charts

State Tables:

You can display a State Machine as a diagram, or as a table in one of three relationship formats.

How to

To select the display format

















Step	Action	See also
1	Right-click on the diagram background and select the Statechart Editor context menu option.	

Step	Action	See also
2	Select the appropriate display option: <ul style="list-style-type: none"> • Diagram • Table (State-Next State) • Table (State-Trigger) • Table (Trigger-State) 	

Example Diagram [Example State Machine Diagram](#) ^[1034]

Tools

Select State Machine elements and connectors from the State pages of the Toolbox; click on the elements and connectors below for more information.

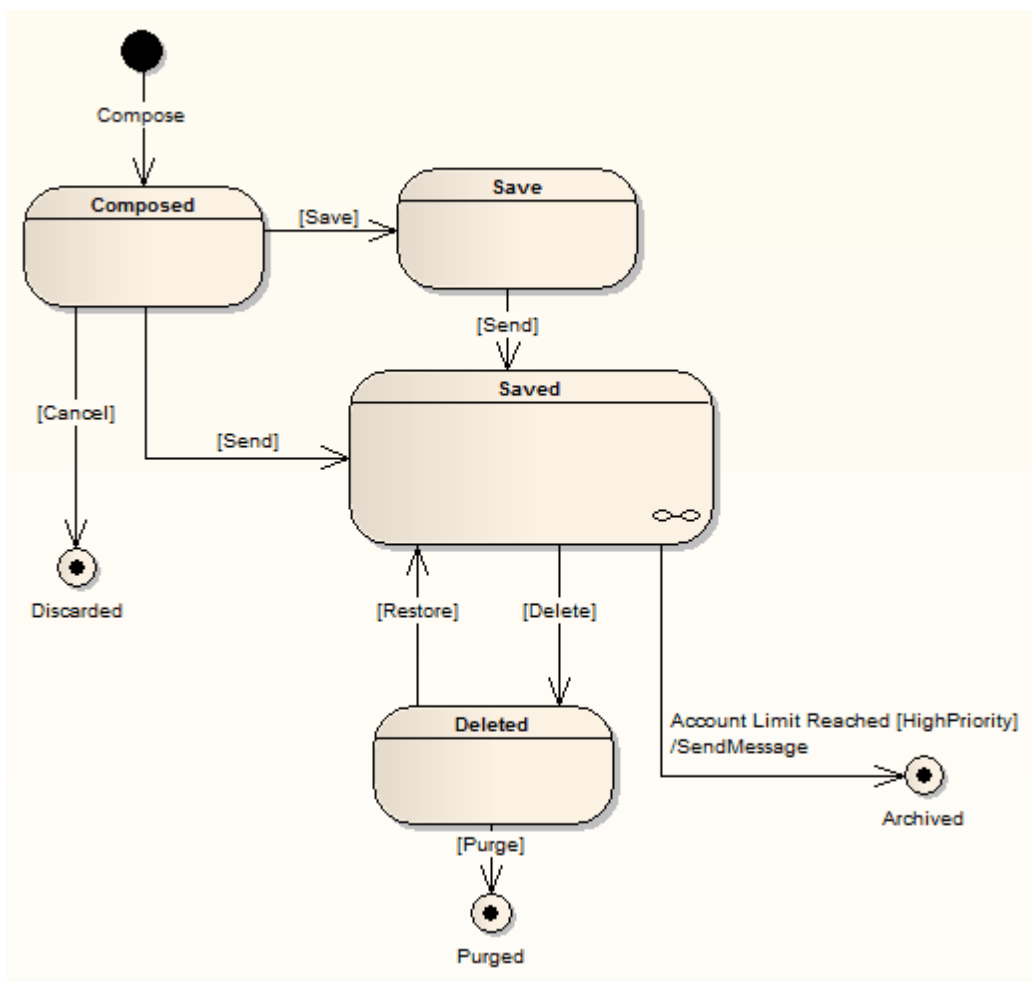
State Machine Diagram Elements	State Machine Diagram Connectors
 State	 Transition
 State Machine	 Object Flow
 Initial	
 Final	
 History	
 Synch	
 Object	
 Choice	
 Junction	
 Entry	
 Exit	
 Terminate	
 Fork/Join	
 Fork/Join	

Learn more

- [SW Code Generation - State Machine Diagrams](#)^[1780]
- [State Machine Modeling for HDLs](#)^[1784]
- [State Machine Table](#)^[1038]
- [State Group](#)^[711]

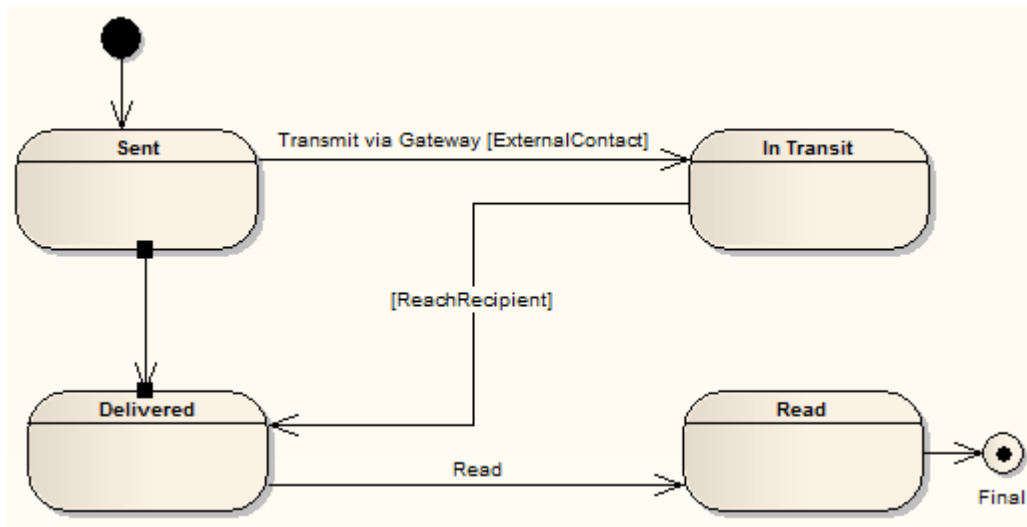
6.3.3.1 Example State Machine

The following diagram illustrates some features of State Machines.

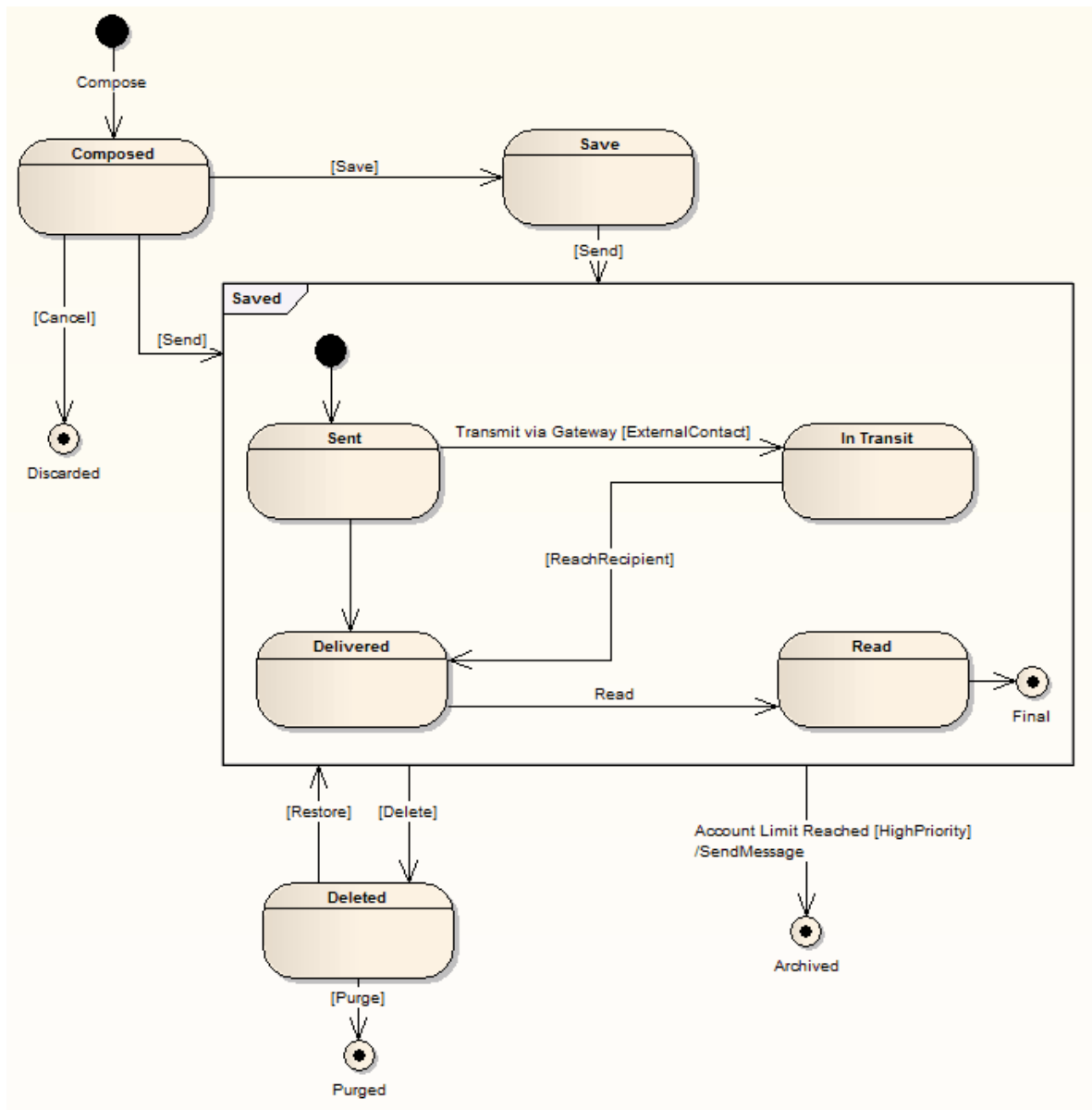
**Composite States:**

The chain-link symbol in the bottom right corner of the *Saved* state indicates that it is a Composite state.

You have two options for exposing the contents of a composite State. Firstly, you can double-click on the element to display its child diagram separately, as shown below:



Alternatively, you can right-click on the composite element and select the **Advanced | Show Composite Diagram** context menu option, which displays the child diagram in the context of the parent diagram.



Notes

- State elements can display either with or without a line across them; the line - as shown above - displays when the element has features such as attributes (which could be hidden) or when the **Show State Compartment** checkbox is selected in the Objects page of the Options dialog



Learn more

- [Show State Compartment](#)^[556]
- [Composite States](#)^[1153]
- [Pseudo-states](#)^[1037]
- [State Elements](#)^[1152]
- [Regions](#)^[1037]

6.3.3.2 Pseudo-States

Pseudo-states are a UML abstraction for various types of transient vertices used in State Machine diagrams. Pseudo-states are used to express complex transition paths.

The following types of pseudo-state are available:

Type	See
 Initial	Initial ^[1136]
 Entry	Entry Point ^[1123]
 Exit	Exit Point ^[1126]
 Choice	Choice ^[1108]
 Junction	Junction ^[1142]
 History	History ^[1134]
 Terminate	Terminate ^[1172]
 Final	Final ^[1127]
 Fork/Join	Fork ^[1131]
 Fork/Join	
 Fork/Join	Join ^[1133]
 Fork/Join	

Learn more

- [State Machines](#) ^[1032]

6.3.3.3 Regions

Regions can be created in Composite States or State Machines on a State Machine diagram. Regions indicate concurrency, such that a single State is active in each region. Multiple transitions can occur from a single event dispatch, so long as similarly triggered transitions are divided by Regions.

How to

To create a Region in a Composite State or State Machine element

Step	Action
1	Right-click on the element, and select the Advanced Define Concurrent Substates context menu option. The State Regions dialog displays.

2	Create the Regions of a State, which can be named or anonymous. Press (Enter) to save each new State Region.
3	Click on the Close button.

Notes

- Changes to the partitions are committed when the diagram is saved; if you want to undo the partition changes, reload the diagram without saving

Learn more

- [Composite States](#) ^[1153]
- [State](#) ^[1152]
- [State Machines](#) ^[1032]

6.3.4 State Machine Table

A State Machine table is one of two variants of a State Machine (the other is the State Machine diagram). It displays the information of the State Machine in table form, and is a method of specifying the discrete behavior of a finite state-transition system; that is, what state the State Machine moves to and the conditions under which the transition takes place.

You can display the state transition as one of two different relationships:

Display Type	Description	See also
State - Trigger	The rows indicate the current states and the columns indicate trigger events (or the other way around if you prefer, in a <i>Trigger - State</i> format). The cell at the intersection of a row and column identifies the target state in the transition if the trigger occurs, and the condition (or guard) of the transition.	Example State-Trigger Table ^[1049]
State - Next State	The rows and columns both indicate states, and the cell at the intersection of a row and column indicates: <ul style="list-style-type: none"> • The event that triggers a transition from the current (row) state to the next (column) state • The condition (or guard) of the event, and • The effect of the transition 	Example State-Next State Table ^[1050]

You can display a State Machine as a diagram or table, and as a table in one of three relationship formats.

How to

To select the display format

Step	Action
1	Right-click on the diagram background and select the Statechart Editor context menu option.
2	Select the appropriate display option: <ul style="list-style-type: none"> • Diagram • Table (State-Next State) • Table (State-Trigger) • Table (Trigger-State)

Learn more

- [State Machine Table Options](#) ^[1039]
- [State Machine Table Operations](#) ^[1042]
- [State Machines](#) ^[1032]

6.3.4.1 State Machine Table Options

You can choose the State Machine table layout and set other options from the State Machine Diagram: Options dialog, which you display by either:

- Double-clicking on the State Machine table background or
- Right-clicking on the background and selecting the **State Table Options** context menu option

Field	Usage	See also
Table Format	Select the required table format: <ul style="list-style-type: none"> • State - Trigger: <ul style="list-style-type: none"> • Rows represent States, each state name in a left edge cell • Columns represent Triggers, each trigger name in a column header cell • The intersection of a row and column identifies the Transition (if there is one) • The Transition cell displays information about the next State and the condition (guard) of the 	

Field	Usage	See also
	<p>Transition</p> <ul style="list-style-type: none"> • Trigger - State: as above, except that rows represent triggers and columns represent states • State - Next State: <ul style="list-style-type: none"> • Both rows and columns represent states • The intersection of row and column defines the transition (if there is one) from the row state to the column state 	
Cell Size		
Transition Cell Width	Specify the width of the transition cells (that is, the column width).	
Transition Cell Height	Specify the height of the transition cells (that is, the row height).	
Left Edge Cell Width	Specify the width of the left edge (row title) cells.	
Top Edge Cell Height	Specify the height of the top edge (column title) cells.	
Cell Color		
State/Trigger Cell	Select the color of the row and column title cells.	
State/Trigger Enumeration	<p>Select the color of the enumeration (row/column numbering) cells.</p> <p>You must select at least one of the Enable State Enumeration and Enable Event Enumeration checkboxes to set this color.</p>	
Transition Cell	Select the color of the transition cells (in the main body of the table).	
Highlight Options		
Highlight Zones Related to Selected Transition	Highlight the cells for all elements involved in a selected transition - the initial state, the target state, and the trigger.	

Field	Usage	See also
Highlight Color	Select the color of the highlight.	
Use Different Color for Target State	Highlight the cell for the target element in a transition in a different color to the cell for the source element.	
Target Zone Color	Select the color of the highlight.	
Display Options		
Always Display an Empty State Zone	Add an empty row (and, on a <i>State - Next State</i> table, an empty column) to the end of the table. The title cell contains an ellipsis (...). You can click twice (not double-click) on the ellipsis to edit it and identify a new state. In this case, another empty state zone is automatically added.	
Enable State Enumeration	Add a cell to each state title cell, to number the state. Numbering starts at 0.	
Prefix	If required, type a prefix for the state number or delete the default S to have no prefix.	
Enable Event Enumeration	Add a cell to each event or trigger title cell, to number the event. Numbering starts at 0.	
Prefix	If required, type a prefix for the event number or delete the default E to have no prefix.	
Sample State Table	Display a preview of the table format as you define it.	
Advanced	Define diagram options. The State Machine Diagram Properties dialog displays.	Set Appearance Options ^[725]
Restore Defaults	Reapply the State Table diagram default values.	
Apply	Apply the changed options to the State Table diagram.	

6.3.4.2 State Machine Table Operations

As a State Machine table is a variant of a State Machine diagram, most of the operations for manipulating the data are the same as for State Machine diagrams. The operations specific to State Machine tables are described in the following topics:

Operation	See also
Change State Machine Table Position	Change Position of State Machine Table ^[1042]
Change State Machine Table Size	Change Size of State Machine Table ^[1043]
Insert New State	Insert New State (and Substate) ^[1043]
Insert Trigger	Insert Trigger ^[1044]
Insert/Change Transition	Insert/Change Transition ^[1045]
Reposition State or Trigger Cells	Reposition State/Trigger Cells ^[1046]
Add Legend	Add Legend ^[1046]
Locate Cell in State Machine Diagram	Find Cell in State Machine Diagram ^[1047]
State Machine Table Conventions	State Machine Table Conventions ^[1047]
Export State Table To CSV File	Export State Table To CSV File ^[1048]

Learn more

- [State Machines](#) ^[1032]
- [State Machine Table](#) ^[1038]

6.3.4.2.1 Change State Machine Table Position

If necessary, you can move the State Machine table around in the Diagram View.

How to

To change the position of the State Machine table

Step	Action
1	Press (Ctrl+A) or double click on the top left cell to select the whole State Machine table.
2	Drag and drop the State Machine table to the required position. Alternatively, use (Shift) + (←) , (↑) , (→) or (↓) to move the State Machine table.

6.3.4.2.2 Change State Machine Table Size

There are three ways to change the size of the State Machine table:

Action	See also
Change the cell size on the State Machine Diagram: Options dialog.	State Machine Table: Options ^[1039]
Press (Ctrl+A) or double click on the top left cell to select the whole State Machine table, then press (Ctrl) + (←) , (→) , (↑) or (↓) to change the size.	
Select the State Machine table, then drag the shape handles to change the size.	

6.3.4.2.3 Insert New State

Topics

Action	Description
Insert a new State in the State Machine table	<p>You can insert a new State in the State Machine table, using one of following methods:</p> <ul style="list-style-type: none"> • In the top left cell in the State Machine table, move the cursor to the word State to display a + at the end of the word; click on the + to create a new State • Right-click in the top left cell in the State Machine table and select the Add State context menu option • Right-click on an existing State cell in the State Machine table and select the: <ul style="list-style-type: none"> • Insert New State Before context menu option to insert a new State before the current State, or • Insert New State After context menu option to insert a new State after the current State • Click on an existing State cell in the State Machine table, and press (Insert) to

	<p>create and insert a new State above the selected State</p> <ul style="list-style-type: none"> In the Toolbox, on the State Elements page, click on an element and then click on: <ul style="list-style-type: none"> The diagram background to add a new State to the end of the table, or An existing State cell to add the new State just above it <p>From the State Elements page of the Toolbox you can insert State, Initial, Final, Entry, Exit and Terminate elements.</p>
Add a Substate to a selected State	<p>To add a Substate to a selected State, right-click on the required State cell in the State Machine table, and select the Add Substate context menu option; Enterprise Architect adds the Substate to the State.</p> <p>If the selected State does not allow a Substate, the Add Substate menu option is grayed out.</p> <p>You can also drag one existing State over another; if the second State allows Substates, the dragged State then becomes its Substate.</p> <p>Similarly, you can change the parent State of a Substate by dragging the Substate from the original parent State to a different State.</p>
Remove the parent relation of a Substate and make it a separate State	<p>To remove the parent relation of a Substate and make it a separate State, right-click on the Substate in the State Machine table and select the Remove Parent Relation context menu option; the Substate cell becomes a State cell.</p> <p>You can also drag and drop the Substate onto the top left cell of the State Machine table; the dragged Substate again becomes a State cell.</p>

6.3.4.2.4 Insert Trigger

If the State Machine table format is either *State-Trigger* or *Trigger-State*, you can use **any** of the following methods to insert a new Trigger:

Method	Action
1	In the top left cell in the State Machine table, move the cursor to the word Event to display a + at the end of the word; click on the + to create a new Trigger.
2	In the top left cell in the State Machine table, right-click and select the Add Trigger context menu option to create a new Trigger.
3	Select an existing Trigger in the State Machine table, then press (Insert) to insert a new Trigger before the existing Trigger.
4	Click on an existing Trigger in the State Machine table, right-click and select either the: <ul style="list-style-type: none"> Insert New Trigger Before context menu option to insert a new Trigger

before the current Trigger, or

- **Insert New Trigger After** context menu option to insert a new Trigger after the current Trigger

6.3.4.2.5 Insert/Change Transition

Action	Description	See also
Insert a new Transition	<p>You can insert a new Transition using one of the following methods:</p> <ul style="list-style-type: none"> • Right-click on the cell in which to create a Transition: <ul style="list-style-type: none"> • If the State Machine table format is <i>State-Trigger</i> or <i>Trigger-State</i>, the context menu lists the States you can choose as the target of the Transition; click on the required State name to create the Transition • If the State Machine table format is <i>State-Next State</i>, click on the Insert Transition context menu option to create the Transition • In the State Relationships page of the Toolbox, select the Transition element, then click on the cell in the State Machine table in which to create the Transition; double-click on the Transition to define it in the Transition Properties dialog 	
Change the Transition	As for the State Chart diagram, to change the properties of a Transition double-click the Transition cell and edit the details on the Transition Properties dialog.	State Machine Diagrams ^[1032]
Change Transition States	<p>You can change the source and target of the Transition by right-clicking the Transition and selecting the Advanced Set Source and Target context menu option.</p> <p>Alternatively, you can change the Transition source, target or Trigger by clicking on the Transition and dragging it to a different cell.</p> <p>If the State Machine table format is either <i>State-Trigger</i> or <i>Trigger-State</i>, you can change the target state of a Transition by:</p> <ol style="list-style-type: none"> 1. Highlighting the target state name in the Transition cell and clicking on it to display a list of the states in the table. 2. Clicking on the preferred target state name. 	
Highlight States and Trigger	You can select options to highlight the source State, target State and Trigger cells associated with a Transition, using the	State Machine Diagram: Options ^[1039]

Related to Transition	<p>Highlight Options panel on the State Machine Diagram: Options dialog.</p> <p>When you click on the Transition cell its associated State and Trigger cells are highlighted.</p> <p>Alternatively, click on the Transition cell and press and hold (L).</p>	
------------------------------	---	--

6.3.4.2.6 Reposition State or Trigger Cells

You can change the position of a selected State or Trigger cell in one of the following ways:

- Right-click on the State or Trigger title cell and select the appropriate **Order | Move xxx** context menu option
- Click on the cell and press **(Shift) + (→) , (←) , (↑)** or **(↓)**

6.3.4.2.7 Add Legend

You can add a simple legend to any State Machine Table cell that has no transition. The two legend symbols are:

- **I** - Ignore
- **N** - Never Happen

How to

To assign a legend symbol to a State Machine Table cell

Step	Action
1	<p>Click on the cell to which to assign the legend and press:</p> <ul style="list-style-type: none"> • (I) to insert the Ignore legend, or • (N) to insert the Never Happen legend <p>The required symbol displays in the center of the cell.</p>

Alternatively:

Step	Action
1	Right-click on the cell to which to assign the legend.
2	<p>Select the appropriate context menu option:</p> <ul style="list-style-type: none"> • Legend Ignore

	<ul style="list-style-type: none"> • Legend Never Happen <p>The required symbol displays in the center of the cell.</p>
--	---

Notes

- To remove a legend symbol from a cell, either:
 - Click on the cell and press (**Delete**), or
 - Right-click on the cell and select the **Legend | Remove Legend** context menu option

6.3.4.2.8 Find Cell in State Machine Diagram

Topics

Locate	Description
In State Chart	<p>On the State Machine table, to locate a selected State or Trigger element in a State Machine diagram:</p> <ul style="list-style-type: none"> • Select the Find Locate in State Chart context menu option <p>Enterprise Architect switches to the State Machine diagram and highlights the selected element.</p> <p>You can locate a Transition relationship in a similar way, by selecting the Locate in State Chart context menu option.</p> <p>A Trigger on a State Machine table might or might not exist on the corresponding State Machine diagram; if the Trigger does not exist on the State Machine diagram, the Locate in State Chart option is disabled.</p>
In State Table	<p>Conversely, on the State Machine diagram, to locate a selected State or Trigger element in the corresponding State Machine table:</p> <ul style="list-style-type: none"> • Select the Find Locate in State Table context menu option <p>Enterprise Architect switches to the State Machine table and highlights the selected element.</p> <p>You can locate a Transition relationship in a similar way, by selecting the Locate in State Table context menu option.</p>

6.3.4.2.9 State Machine Table Conventions

Trigger:

- Deleting a Trigger removes it completely from the model, therefore you cannot UNDO a Trigger

deletion

- There is a <None> column at the end of the **Event** heading row; this is for Transitions that have no Trigger information

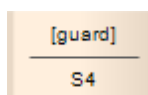
State:

From the Toolbox you can insert the following *State* element types only (although the State Machine table might pick up and display other types, such as *Submachine State*):

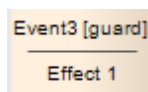
- State
- Initial
- Final
- Entry
- Exit
- Terminate

Transition:

The Transition cell displays its properties in one of two ways, depending on the State Machine table format.; if the State Machine table format is *State - Trigger* or *Trigger - State*, the Transition cell displays the *Guard* and *Target* as shown below:



If the State Machine table format is *State - Next State*, then the Transition cell displays the *Trigger*, *Guard* and *Effect* as shown below:



The State Machine table enables you to edit the *Guard* and *Effect* in place. If the *Guard* or *Effect* is empty for your selected Transition cell, the cell displays an ellipsis (...) instead; click twice (not double-click) on the ellipsis to type in the *Guard* and *Effect* names.

6.3.4.2.10 Export State Table To CSV File

To export a State Machine Table to a CSV file

Step	Action
1	Open the required State Machine Table.

2	Right-click on the diagram background and select the Export Statechart to CSV file context menu option. The Save As browser dialog displays.
3	Select the appropriate directory location and type in the .CSV filename.
4	Click on the Save button.

6.3.4.3 Example State-Trigger Table

State - Trigger:

The rows indicate the current states and the columns indicate trigger events (or the other way around if you prefer, in a **Trigger - State** format).

The cell at the intersection of a row and column identifies the target state in the transition if the trigger occurs, and the condition (or guard) of the transition.

State \ Trigger		Event1	Event2	Event3	Event4	<None>
		E0	E1	E2	E3	E4
Initial	S0					S1
State1	S1				S2	
State2	S2	S6	<u>[Guard]</u> S4			
	SubState1	S3	S4			
	SubState2	S4		<u>[Cond]</u> S2		
	SubState3	S5				
State3	S6					S7
Final	S7					

6.3.4.4 Example State-Next State Table

State - Next State:

The rows and columns both indicate states, and the cell at the intersection of a row and column indicates:

- The event that triggers a transition from the current (row) state to the next (column) state
- The condition (or guard) of the event, and
- The effect of the transition



6.3.5 Timing Diagram

A Timing diagram defines the behavior of different objects within a time-scale. It provides a visual representation of objects changing state and interacting over time.

Use to







- Define hardware-driven or embedded software components; for example, those used in a fuel injection system or a microwave controller
- Specify time-driven business processes

Example Diagram [Example Timing Diagram](#) ¹⁰⁵²

Tools

Select Timing diagram elements and connectors from the Timing pages of the Toolbox.

Click on the elements and connectors below for more information.

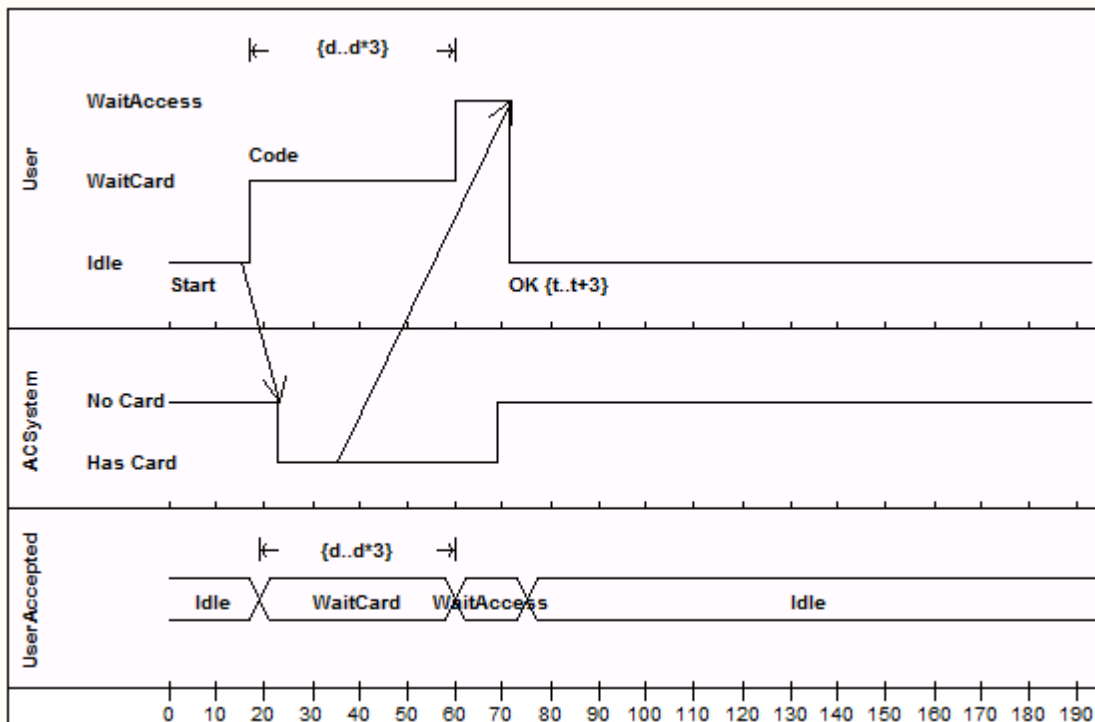
Timing Diagram Elements	Timing Diagram Message
 State Lifeline	 Message
 Value Lifeline	
 Message Label	
 Message Endpoint	
 Diagram Gate	

Learn more

- [Create a Timing Diagram](#)^[1053]
- [Set a Time Range](#)^[1053]
- [State Lifeline](#)^[1158]
- [Value Lifeline](#)^[1177]
- [Edit a Timing Diagram](#)^[1054]
- [Time Intervals](#)^[1068]
- [Message \(Timing Diagram\)](#)^[1245]
- [Timing pages](#)^[71†]

6.3.5.1 Example Timing Diagram

An example of a Timing diagram is shown below:



6.3.5.2 Create a Timing Diagram

How to

To create a Timing diagram

Step	Action
1	Right-click on a package in the Project Browser and select the Add Add Diagram context menu option. The New Diagram dialog displays.
2	In the Select From panel, select UML Behavioral .
3	In the Diagram Types panel, select Timing .
4	Click on the OK button. The Diagram View displays, on which you create the Timing elements for the diagram.

Learn more

- [Set a Time Range](#)^[1053]
- [Edit a Timing Diagram](#)^[1054]

6.3.5.3 Set a Time Range

How to

To set a time range before adding Lifeline elements to your Timing diagram

Step	Action
1	Right-click on the diagram and select the Set Timeline Range context menu option. The Set Timeline Range dialog displays.
2	In the Start Time and End Time fields, type the numeric values for the start and end points of the timeline; for example, set the range 0 to 100 . The start time must be less than the end time.
3	In the Time Units field, type the unit in which the time is measured; for example, seconds or minutes.

4	If it is not necessary to show the time range on the diagram, select the Suppress In Diagram checkbox.
5	Click on the OK button. If you have not suppressed it, the time range displays underneath the Lifeline elements that you create on the diagram.

6.3.5.4 Edit a Timing Diagram

On a Timing Diagram, you can add State Lifeline elements and Value Lifeline elements. You can maintain the states and transitions on these Lifeline elements either on the diagram itself or via the Configure Timeline dialog.

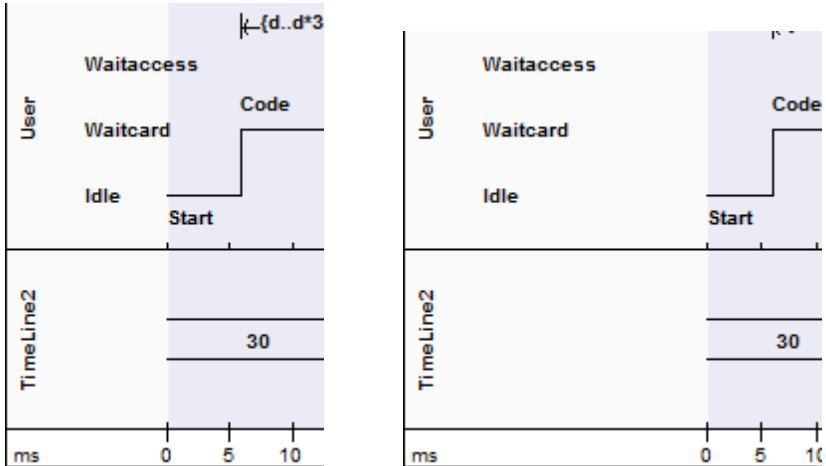
Learn more

- [Add and Edit a State Lifeline Element](#)^[1054]
- [Add States to a State Lifeline](#)^[1055]
- [Edit States in a State Lifeline](#)^[1056]
- [Delete States in a State Lifeline](#)^[1057]
- [Edit Transitions in a State Lifeline Element](#)^[1057]
- [Add and Edit a Value Lifeline Element](#)^[1060]
- [Add States in a Value Lifeline Element](#)^[1061]
- [Edit Transitions in a Value Lifeline Element](#)^[1061]
- [Configure Timeline dialog - States Tab](#)^[1062]
- [Configure Timeline dialog - Transitions Tab](#)^[1065]

6.3.5.4.1 Add and Edit State Lifeline

From the Timing elements page of the Toolbox drag a State Lifeline icon onto your diagram. The element displays on the diagram.

Task	Action
Define the name of the State Lifeline	<ol style="list-style-type: none"> 1. Right-click on the element and select the Other Properties context menu option; the Timeline <name> dialog displays, showing the General tab. 2. Overtyping the Name field. 3. Click on the Apply button and the OK button.
Sizing and Scale	<p>In the top left corner of a selected Lifeline element are the left and right quick sizing buttons (← →).</p> <p>These buttons increase or decrease the width of the Lifeline element, which in turn controls the scale width of each time unit; by increasing the width of the element you increase the resolution when adding transitions, which makes them easier to edit.</p>

	In order to edit the State Lifeline element, you must click on it to select it.
Set Timeline Start Position	<p>You might require more space at the start of your timelines; for example, to use long state names.</p> <p>To insert more space in all the timelines on a diagram</p> <ol style="list-style-type: none"> 1. Right-click on the diagram background and select the Set Timeline Start Position menu option; the Set Timeline Start Position dialog displays. 2. The Value 80 to 300 field defaults to 80 as the minimum distance in pixels between the start of the timeline element and the start of the timeline itself; type a new value up to 300 pixels and click on the OK button to increase the space at the start of the timeline, as shown in the following diagrams (80 pixels and 150 pixels respectively).  <p>You now edit the states and transitions in the State Lifeline.</p>

Learn more




- [Add States to a State Lifeline](#) ^{f1055}
- [Edit States in a State Lifeline](#) ^{f1056}
- [Delete States in a State Lifeline](#) ^{f1057}
- [Edit Transitions In State Lifeline](#) ^{f1057}

6.3.5.4.2 Add States to a State Lifeline

How to

To add States to a State Lifeline



Step	Description	See also
------	-------------	----------

1	Click on the <i>State Lifeline</i> element. The New State button () and Edit States button () display at the bottom left of the element.	
2	Click on the New State button. The New State dialog displays.	
3	In the State field, type the name of the state.	
4	Click on the OK button. You must add at least two states; for example, On and Off .	
5	As you add states, increase the height of the element by dragging a handle-box () on the edge of the element. You can also add states using the States tab of the Configure Timeline dialog. Add either: <ul style="list-style-type: none"> • Discrete states to the Timeline, or • A continuous range of numeric states 	Add a New State <small>[1063]</small> Numeric Range Generator <small>[1064]</small>

6.3.5.4.3 Edit States in a State Lifeline

How to

To edit States in a State Lifeline

Step	Description	See also
1	Click on the State Lifeline element and click on the required state. The Edit State dialog displays.	
2	In the State field, change the name as required.	
3	Click on the OK button.	
4	If necessary, change the order of the states by either: <ul style="list-style-type: none"> • Clicking on the up or down arrows ( and ) beside each state name, or • Right-clicking on the state name and selecting the Move Up or Move Down context menu options 	States <small>[1062]</small>

	You can also edit the states using the States tab of the Configure Timeline dialog.	
--	---	--

6.3.5.4.4 Delete States in a State Lifeline

How to

To delete States in a State Lifeline

Step	Description
1	Right-click on the state name and select the Delete context menu option.

Alternatively:

Step	Description
1	Click on the State Lifeline element.
2	Hold down (Ctrl) and move the cursor over the state name. The cursor changes form (↔).
3	Click the mouse button. The state name is deleted.

6.3.5.4.5 Edit Transitions In State Lifeline

A Timing diagram allows you to show the transitions (changes of state) that occur within a state machine over a fixed time period and at certain timing points. This is similar in many respects to an Interaction lifeline with State changes highlighted. As events and changes occur within the instance this Timing diagram represents, state changes occur and are mapped onto this Timeline. In that respect it is a record of how a particular aspect of the system behaves over time.

When building a Timeline it is necessary to define the States first - and then to add the explicit transitions between those states at particular timing points.

Task	Action	See also
Add and Move Transitions	After you have added states, you can add transitions between states directly on the timeline using the mouse.	Add and Move Transitions ^[1058]



Change the Transition Time	<p>Move the cursor over one or other of the vertical transition lines and drag the line left or right to change the time of the transition.</p> <p>While on the line, the cursor shape changes to the horizontal movement cursor (↔).</p>	
Merge Transitions	<p>If necessary, you can 'push' a transition to merge it with the next or previous transition point on any Lifeline element on the diagram.</p> <p>Position the cursor off the appropriate side of the transition line; the cursor changes form (← or →).</p> <p>Click the mouse button; the system locates the nearest transition in the required direction, on any element on the diagram, and merges the current transition with that transition.</p>	
Delete Transitions	<p>Transitions are automatically deleted when you move the transition to the same state as the previous transition state, and release the cursor.</p> <p>Alternatively, right-click on the transition line and select the Delete context menu option.</p>	

6.3.5.4.5.1 Add and Move Transitions

After you have added states, you can configure state changes (transitions) directly on the Timeline using the mouse. This is a fast and effective means of building a detailed model of state changes over time.

In order to modify the Timeline, place the mouse over the existing Timeline. As you move the cursor over the Timeline, the cursor changes to one of three shapes, described below.

Task	Action	See also
The move cursor (↕)	<p>Displays when it is directly over the timeline.</p> <p>Hold down the mouse button and drag the line to move the timeline to a state above or below the current position; you can move the transition more than one state up or down, if necessary.</p>	
The new transition up cursor (↕)	<p>Displays when it is just below the timeline, and there is another state above the line.</p> <p>Press and hold (Alt); the cursor changes (↕).</p> <p>Click to create a new transition to the state above the line.</p> <p>To push the transition up more than one state, move the cursor onto the line and drag it up.</p> <p>The transition is for one interval unit; you can make it longer by changing the transition time.</p>	Edit Transitions In State Lifeline ^[1057]

	If you do not hold (Alt), the cursor does not change and the whole timeline from the transition onwards moves up.	
The new transition down cursor ()	<p>Displays when it is just above the transition line, and there is another state below the line.</p> <p>Press and hold (Alt); the cursor changes ().</p> <p>Click to create a new transition to the state below the line.</p> <p>To push the transition down more than one state, move the cursor onto the line and drag it down.</p> <p>The transition is for one interval unit; you can make it longer by changing the transition time.</p> <p>If you do not hold (Alt), the cursor does not change and the whole timeline from the transition onwards moves down.</p>	Edit Transitions In State Lifeline ^[1057]

As you move the cursor over the vertical line of a transition, the time at which the transition occurs displays next to the line.

Access Click directly on the appropriate transition line, after the transition begins, or Right-click on the transition line and select the **Edit** context menu option

Edit the transitions as required, on the Edit Transition dialog.

Field/Option	Action
At Time	Type the point on the timescale at which the transition occurs.
Transition To	Type the name of the state to which the transition occurs.
Event	Type the name of the event that the transition represents. This displays on the Timeline element just above the transition line.
Duration Constraint	Type any constraint on the duration of the transition. This displays on the Timeline element, along the top of the element over the transition.
Time Constraint	Type any constraint on the start of the transition. This displays on the Timeline element at the start of the transition.

OK	Click on this button to save the changes.
-----------	---

Notes

- Once Event, Duration Constraint or Time Constraint are displayed on the diagram, you can edit them directly by clicking on them to display their specific dialog
- You can delete them by pressing and holding (**Ctrl**) as you click on them; the cursor changes form when you press (**Ctrl**)
- You can also edit transitions using the Transitions tab of the Configure Timeline dialog

Learn more

- [Transitions](#) 

6.3.5.4.6 Add and Edit Value Lifeline



From the Toolbox drag a Value Lifeline element onto your diagram. The element displays on the diagram.

How to




To edit the Value Lifeline name

Step	Action
1	Right-click on the element and select the Other Properties context menu option. The Timeline <name> dialog displays, showing the General tab.
2	Overtyping the Name field.
3	Click on the Apply button and the OK button.

Sizing and Scale:

In the top left corner of a selected Lifeline element are the left and right *quick sizing* buttons ( ). These buttons increase or decrease the width of the Lifeline element, which in turn controls the scale width of each time unit. By increasing the width of the element you increase the resolution when adding transitions, which makes them easier to edit.

Learn more

- [Value Lifeline](#) 
- [Add States in a Value Lifeline](#) 
- [Edit Transitions in a Value Lifeline](#) 

6.3.5.4.7 Add States In Value Lifeline

Adding States to a Value Lifeline is similar to adding states to a State Lifeline element.

For a Value Lifeline, only the first state displays on the diagram. The other states are added to a list to access when creating transitions; they only display on the Lifeline element as you create transitions to those states.

You can only edit or delete states in a Value Lifeline element using the States tab of the Configure Timeline dialog.


Learn more

- [Add States to a State Lifeline](#) 

6.3.5.4.8 Edit Transitions In Value Lifeline

How to

To add transitions to the states on a Value Lifeline element, via the diagram

Step	Action
1	Move the cursor above the transition line. The cursor changes form ().
2	Click the mouse button. The New Transition Event dialog displays.
3	In the Transition To field, click on the drop-down arrow and select a state from the list of available states; this displays on the Lifeline element within the transition box. The remaining fields on the dialog are optional.
4	In the Event field, type the name of the event that the transition represents; this displays on the Lifeline element just below and at the start of the transition line.
5	In the Duration Constraint field, type any constraint on the duration of the transition; this displays on the Lifeline element, along the top of the element over the transition.
6	In the Time Constraint field, type any constraint on the start of the transition. This displays on the Lifeline element at the start of the transition, just after the Event name.
7	Click on the OK button to create the new transition.

To edit a transition

Step	Action
1	Click on the state name in the transition. Alternatively, right-click on the state name and select the Edit context menu option. The Edit Transition dialog displays, which is the same as the New Transition Event dialog, except that the At Time field is enabled.
2	If necessary, overwrite the At Time field to define a different start point. You cannot change the At Time field for the first state in the timeline; this is always 0 .
3	Edit the remaining fields as necessary.
4	Click on the OK button to save the changes.

To change the transition time

Step	Action
1	To change the start or end time of a transition, click on the start or end point of the transition and drag it to the new position. While on the line, the cursor shape changes to the horizontal movement cursor (↔).


To delete Transitions

Step	Action
1	To delete a transition, press and hold (Ctrl) and click on the transition state name. While you hold (Ctrl) on the transition state name, the cursor changes form (↔). Alternatively, right-click on the state name and select the Delete context menu option.

6.3.5.4.9 Configure Timeline - States

You can manage states using the States tab of the Configure Timeline dialog. To display this, either:

- Double-click on the Lifeline element
- Right click on the Lifeline element and select the **Properties** context menu option, or

- On a Value Lifeline, click on the **Edit States** button ()

The Configure Timeline dialog defaults to the States tab.

All states currently defined for the Lifeline element are listed in the States panel.

How to

To add a new State

Step	Action
1	In the State Name field, type the name of the first new state in the Lifeline element; for example, WaitState .
2	Click on the Save button. The state is added to the States panel and (for a State Lifeline Element) to the diagram.
3	Click on the New button.
4	In the State Name field, type the name of the next state in the Lifeline element.
5	Repeat steps 2 to 5 until you have added all required states (you must add at least three to the Lifeline element).
6	When you have added all the required states, click on the OK button to close the Configure Timeline dialog.



To edit an existing state

Step	Action
1	Click on the state in the States: list.
2	In the State Name field, change the name of the state.
3	Click on the Save button.

To delete an existing State

Step	Action
1	Click on the state in the States: list.
2	Click on the Delete button.

To change the order of States

Step	Action
1	Click on the state in the States: list.
2	Click on the  or  buttons to move the state up or down the sequence.

See also:

- [Numeric Range Generator](#)^[1064]

6.3.5.4.10 Numeric Range Generator

You can also use the Configure Timeline dialog to create a range of states having numeric values to be applied to the Timeline.

Important: This operation deletes all existing states and transitions for the Timeline element.

How to


To create a range of states having numeric values

Step	Action
1	Double-click on the Lifeline element. The Configure Timeline dialog displays.
2	Click on the Create Continuous Numeric States button. The Numeric Range Generator dialog displays.
3	In the High Value and Low Value fields, type the upper and lower values of the range.
4	In the Step Value field, type the increase interval.

	Nonsense values do not parse; Low Value must be less than High Value , and Step Value must be a positive value smaller than the total range.
5	In the Units field, type the name of the measurement unit; for example, minutes .
6	Click on the OK button. Enterprise Architect displays a warning that existing states and transitions are to be deleted.
7	Click on the Yes button. The Configure Timeline dialog redisplay, with the defined range of states listed in the States panel.
8	Click on the OK button. For a: <ul style="list-style-type: none"> • Value Lifeline, the first state is shown on the Timeline for the full time range of the Timeline • State Lifeline, the range of states is displayed as the y-axis of the Timeline

6.3.5.4.11 Configure Timeline - Transitions

You can also manage transitions using the Transitions tab of the Configure Timeline dialog. To display this, either:

- Double-click on the Lifeline element
- Right click on the Lifeline element and select the **Properties** context menu option, or
- On a Value Lifeline, click on the **Edit States** button ()

The Configure Timeline dialog defaults to the States tab. Click on the Transitions tab.

All transitions defined for the Timeline element are listed in the Transition Points panel.

How to

To add a new transition

Step	Action
1	Click on the New button.
2	In the New Transition panel, type the details of the transition.

3	Click on the Save button.
----------	----------------------------------

To edit a transition

Step	Action
1	Click on a transition in the list.
2	In the Edit Transition panel, edit the fields for the transition as required.
3	Click on the Save button.

To delete a transition

Step	Action
1	Click on a transition in the list.
2	Click on the Delete button. The transition is removed from the dialog and the Lifeline.
3	Click on the OK button.

6.3.5.5 Time Intervals

You create and manage Time Intervals using the *Interval Bar* (the pale line along the top of each selected Lifeline element). Time Intervals enable you to perform various operations on transitions, such as copy and paste. They also enable you to compress sections of the timeline so that they are not visible.

Each Time Interval displays across all Timeline elements down to the last element on the diagram.


Action	Description	See
Create Time Intervals	You can create a Time Interval using the: <ul style="list-style-type: none"> Interval Bar - Context menu Interval Bar - (Shift) key, or Timeline - Context menu 	Create Time Intervals <small>[1067]</small>
Compress	You can compress Time Intervals to conserve space on long	Compress Time

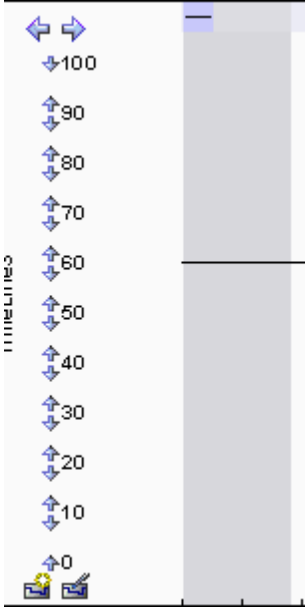
Time Intervals	timelines.	Intervals ^[1069]
Select Time Intervals	There are a number of ways to select Time Intervals for performing other operations.	Select Time Intervals ^[1070]
Move Time Intervals	To move a Time Interval, move the cursor over the Interval bar within the Time Interval, hold down the mouse button and drag the interval left or right. Time Intervals can meet, but cannot overlap.	
Resize Time Intervals	To resize a Time Interval, move the cursor over the Interval Bar at the start or end edge of the Time Interval, hold down the mouse button and move the edge left or right. Time Intervals can meet, but cannot overlap.	
Delete Time Intervals	To Delete Time Intervals, select each Time Interval to be deleted and press (Delete). Deleting the Time Interval does not delete transitions within that interval.	

6.3.5.5.1 Create Time Intervals

How to

To create a Time Interval using the Interval Bar context menu

Step	Description	
1	Right-click on the Interval Bar at approximately the point at which to start or finish the Time Interval. The context menu displays.	

2	<p>Select the Create Time Interval option.</p> <p>The Time Interval displays down all the timeline elements, as a narrow pale band with a blue compression box at the top.</p>	
3	<p>Move the cursor to the edge of the Time Interval in the Interval Bar so that the cursor changes to the drag form (↔) and drag the edge to the correct start or end point.</p>	

To create a Time Interval using the Interval Bar and (**Shift**) key

Step	Description	See also
1	<p>Move the cursor over the Interval Bar and press (Shift).</p> <p>The cursor changes shape (↔).</p>	
2	<p>Click to create the Time Interval.</p>	
3	<p>Move the cursor to the edge of the Time Interval in the Interval Bar so that the cursor changes to the drag form (↔) and drag the edge to the correct start or end point.</p>	

To create a Time Interval using the Timeline context menu

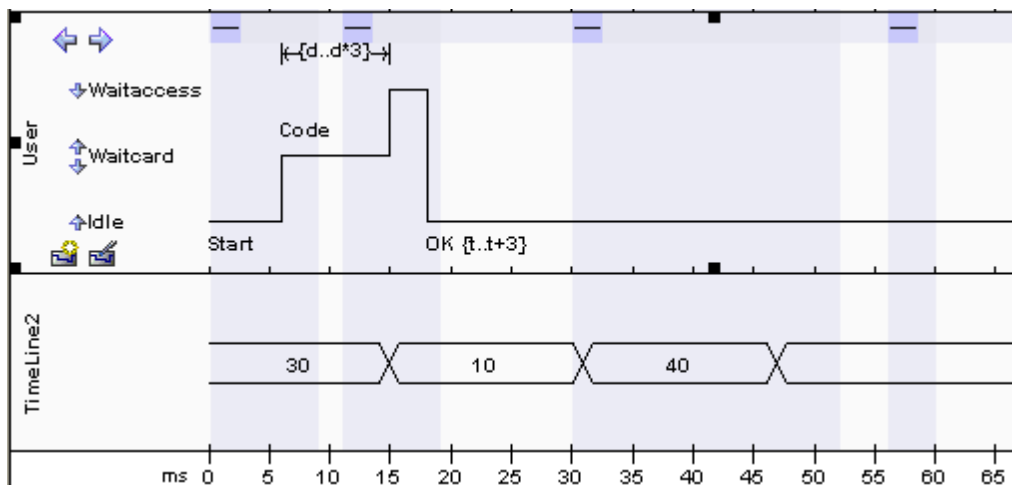
Step	Description	See also
1	<p>Right-click on the timeline just after a transition.</p>	

	The context menu displays.	
2	<p>Click on the Select menu option.</p> <p>Enterprise Architect creates a Time Interval covering the period from the selected transition up to the next transition.</p> <p>If there are other Time Intervals in this period, Enterprise Architect replaces them with the single Time Interval for the transition state; you should consider this when creating the Time Interval, as it extends across the other Timeline elements in the diagram.</p> <p>A value of this method is that it creates a Time Interval for a period in which no transitions occur, which could be lengthy; you can then compress this Time Interval to hide the period of inactivity.</p>	Compress Timeline ^[1072]

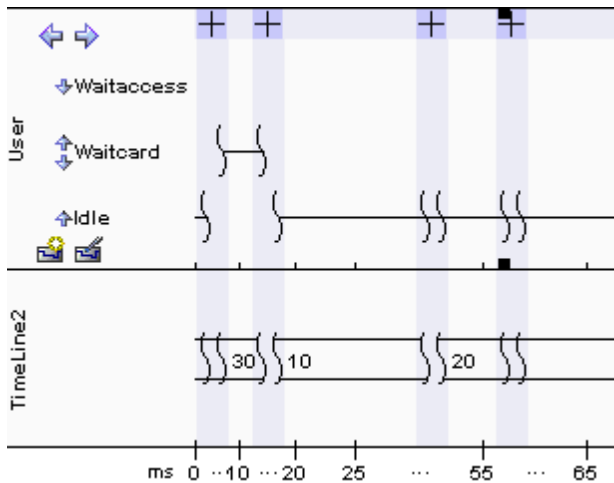
6.3.5.5.2 Compress Time Intervals

You can compress Time Intervals to conserve space on long timelines.

Uncompressed Time Intervals:



Compressed Time Intervals:



Notice:

Item	Description
	<p>The compression toggle boxes:</p> <ul style="list-style-type: none"> Expanded (minus) is expanded, click on this to compress the selected time interval Compressed (plus) is compressed, click on this to expand the selected time interval again
	<p>The compressed sections of the timelines themselves, in all elements.</p> <p>If there is space between the paired symbols, there are transitions within the compressed section.</p> <p>If the timeline continues through the paired symbols there are no transitions in the compressed section.</p>
25 ... 55	The compressed sections in the time range underneath the elements.


You can also compress and expand Time Intervals using context menu options; see [Time Interval Operations on Transitions](#)^[1071].

6.3.5.5.3 Select Time Intervals

Task	Action	See also
Select a Time Interval across all elements on the diagram	Click on the Interval Bar within the Time Interval.	
Select a number of individual Time Intervals	Press and hold (Ctrl) while clicking on the Interval Bar within each Time Interval.	

Select all Time Intervals in a range	Click on the Interval Bar within the first Time Interval in the range, then press and hold (Shift) and click on the Interval Bar within the last Time Interval in the range. All Time Intervals between the two are selected.	
---	---	--




After you have selected one or more Time Intervals, you can modify the selection in the following ways:

Task	Action	See also
Exclude Lifeline elements from the selection	Press and hold (Ctrl) and click on any part of the selection within that element. Repeat the step to toggle the selection and re-include the element.	Toggle Interval Selection 
Select only one Lifeline element and exclude all others	Press and hold (Shift) and click on any part of the selection within that element.	

6.3.5.5.4 Time Interval Operations

This topic discusses how you select and update specific Time Intervals.

Right-click on the Interval Bar within an interval. A context menu displays providing the following options:

Option	Action	See also
Select Interval Deselect Interval	Select the Time Interval or, if the interval is already selected, deselect it. You can select several Time Intervals in this way, accessing the menu separately on each interval.	Select Time Intervals 
Toggle Interval Selection	Switch the selection or deselection of the Time Interval within the selected Timeline element. You select or deselect a Time Interval across all Timeline elements, but the Toggle option acts only on the element in which you access the menu.	Select Time Intervals 
Compress Interval	Compress the Time Interval, and hide all transitions within that Time Interval. This is also useful for hiding long sections of inactivity on the time line.	Compress Timeline 
Remove Interval	Delete the Time Interval.	

Option	Action	See also
Copy	Copy the transitions for all selected Time Intervals.	
Cut	Copy and delete the selected transitions from the diagram.	
Cut and Remove Time	Copy and delete the transitions that lie in the selected Time Intervals from the diagram. This option also removes time from the timeline, the amount being the duration of the Time Interval. All transitions and Time Intervals to the right of the selected time interval are moved left.	
Delete	Delete the selected transitions from the diagram.	
Delete and Remove Time	Delete the transitions that lie in the selected Time Intervals from the diagram. This option also removes time from the timeline, the amount being the duration of the Time Interval. All transitions and Time Intervals to the right of the current Time Interval are moved left.	
Insert Time	Add time to the timeline and move all transitions and time intervals to the right. Also expand the duration of the current Time Interval.	

Compress Timeline:

The Compression toggle boxes and **Compress Interval** menu option operate on the Time Interval and compress the timeline and all transitions within the Interval. You have an alternative option that operates on the timeline and compresses a single transition state.

1. Right-click on the timeline (rather than the Interval Bar) just after a transition, and select the **Compress** context menu option.
2. Enterprise Architect creates a new Time Interval covering the period from the selected transition up to the next transition, and then compresses that Time Interval.

If there are other Time Intervals in this period, Enterprise Architect replaces them with the single Time Interval for the transition state. You should consider this when creating and compressing the Time Interval, as it extends across the other Timeline elements in the diagram.

A value of this method is that it creates a Time Interval for a period in which no transitions occur, which could be lengthy, and then compresses this Time Interval to hide the period of inactivity.

All Time Intervals in the Diagram:

To create a new Time Interval or work across all Time Intervals in the diagram, right-click on the Interval Bar between Time Intervals. A context menu displays, providing the following options (The **Paste ...** menu options become active after transitions have been copied).

Menu Option	Action	See also
Create Time Interval	Create a single Time Interval.	Create Time Intervals [1067]
Expand all Time Intervals	Expand all Time Intervals over the whole diagram.	
Compress all Time Intervals	Compress all Time Intervals over the whole diagram.	
Paste Combine	Paste copied transitions over any existing transitions within the copied time frame. The diagram does not allow two consecutive transitions to the same state, and removes the second transition automatically.	
Paste Remove	Delete all the transitions and then pastes the copied transition within the copied time frame.	
Paste Insert	Insert time, moving all transitions and Time Intervals to the right to make room to paste in the copied transitions.	
Insert Time	Add time to the timeline and move all transitions and Time Intervals to the right. This option does not change the duration of any Time Interval.	

To copy and paste transitions from one timeline element to another

Step	Action	See also
1	Press and hold (Shift) and select the Timeline element within a Time Interval to copy or cut.	
2	Right-click on the Interval Bar (it doesn't matter which element you select). The context menu displays.	
3	Copy or cut the transitions. You can also cut and remove time.	Copy or cut [1072]

Step	Action	See also
4	Select the timeline to paste transitions to and right-click on the Interval Bar. The context menu displays.	
5	Select one of the paste operations. Note that states are created if they don't already exist in the timeline. Any states that don't exist in the Timeline element you are pasting to are created. Any new states created might be in the wrong order; you can change the order via the diagram quick buttons.	Quick Buttons ^[1054]

To shift transitions within a selected Time Interval or multiple selected Time Intervals

Step	Action	See also
1	Select all the Time Intervals containing the transitions to be shifted.	Select Time Intervals ^[1070]
2	Press and hold (Shift) and click on the Interval Bar (it doesn't matter which Timeline element you select), and move the transition left or right. You cannot drag transitions over other transitions; the move stops when the moved transition collides with a stationary transition. If you have collision problems, use (Shift) + select to shift transitions for a single Timeline element.	

6.3.6 Sequence Diagram

A Sequence diagram is a structured representation of behavior as a series of sequential steps over time.

Use to

- Depict work flow, message passing and how elements in general cooperate over time to achieve a result
- Capture the flow of information and responsibility throughout the system, early in analysis; messages between elements eventually become method calls in the Class model
- Make explanatory models for Use Case scenarios; by creating a Sequence diagram with an Actor and elements involved in the Use Case, you can model the sequence of steps the user and the system undertake to complete the required tasks

Construction:

- Each sequence element is arranged in a horizontal sequence, with messages passing back and forward between elements

- Messages on a Sequence diagram can be of several types; the Messages can also be configured to reflect the operations and properties of the source and target elements (see the Notes in the *Message* topic)
- An Actor element can be used to represent the user initiating the flow of events
- Stereotyped elements, such as Boundary, Control and Entity, can be used to illustrate screens, controllers and database items, respectively
- Each element has a dashed stem called a Lifeline, where that element exists and potentially takes part in the interactions














To toggle the numbering of messages on a Sequence diagram, select or deselect the **Show Sequence Numbering** checkbox on the Options dialog.

Example Diagram [Example Sequence Diagram](#)^[1076]

Tools

Select Sequence diagram elements and connectors from the [Interaction pages](#)^[710] of the Toolbox.

Click on the following elements and connectors for more information.

Sequence Diagram Elements	Sequence Diagram Connectors
 Actor	 Message
 Lifeline	 Self-Message
 Boundary	 Recursion
 Control	 Call
 Entity	
 Fragment	
 Endpoint	
 Diagram Gate	
 State/Continuation	

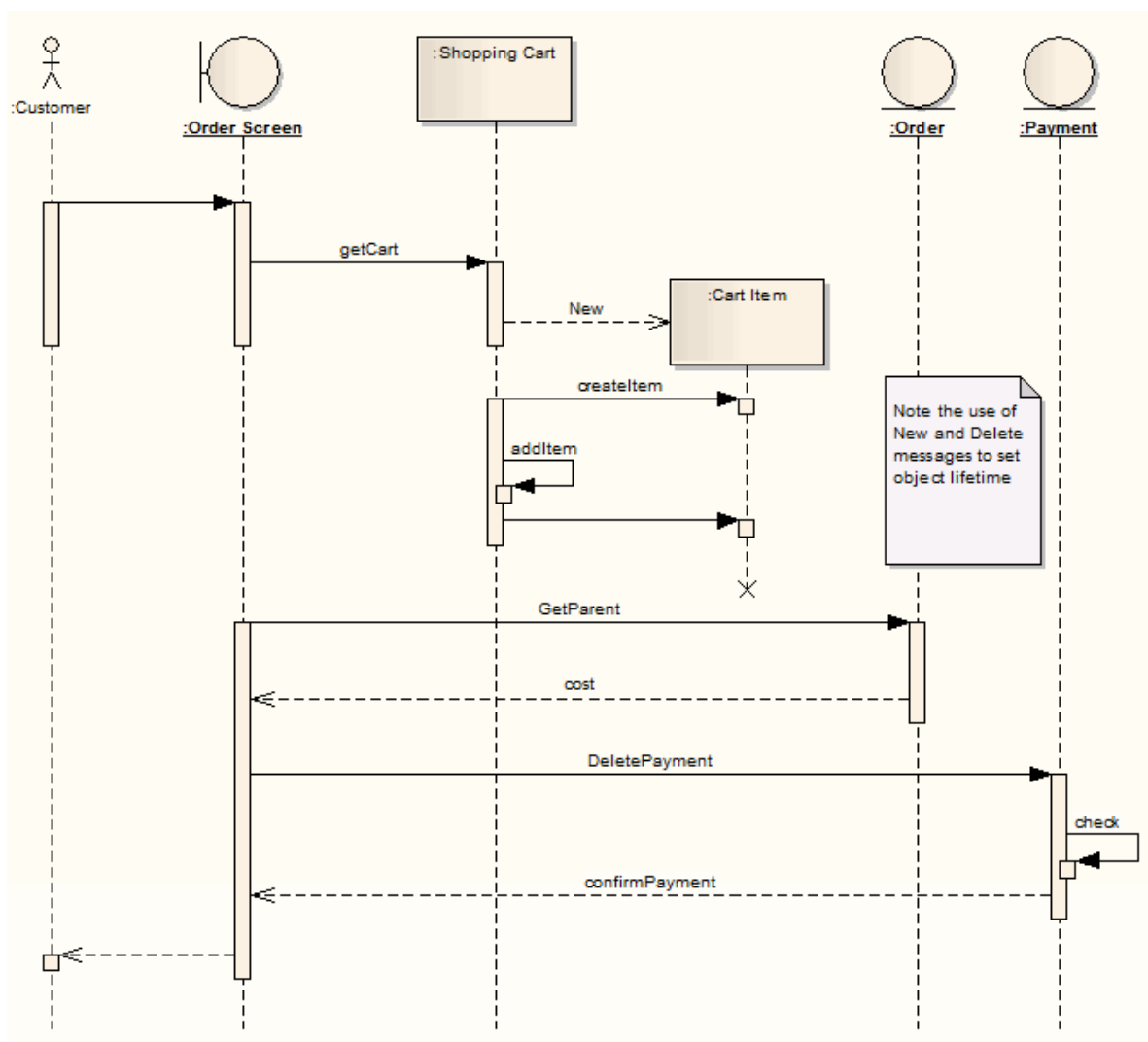
Learn more

- [Denote the Lifecycle of an Element](#)^[1076]
- [Layout of Sequence Diagrams](#)^[1077]
- [Sequence Element Activation](#)^[1080]

- [Lifeline Activation Levels](#)^[1082]
- [Message Label Visibility](#)^[1083]
- [Change the Top Margin](#)^[1083]
- [Change the Timing Details](#)^[1238]
- [Business Modeling/Interaction](#)^[1514]
- [Sequence Diagrams and Version Control](#)^[1079]
- [Show Sequence Numbering](#)^[555]

6.3.6.1 Example Sequence Diagram

The following example Sequence diagram demonstrates several different elements:



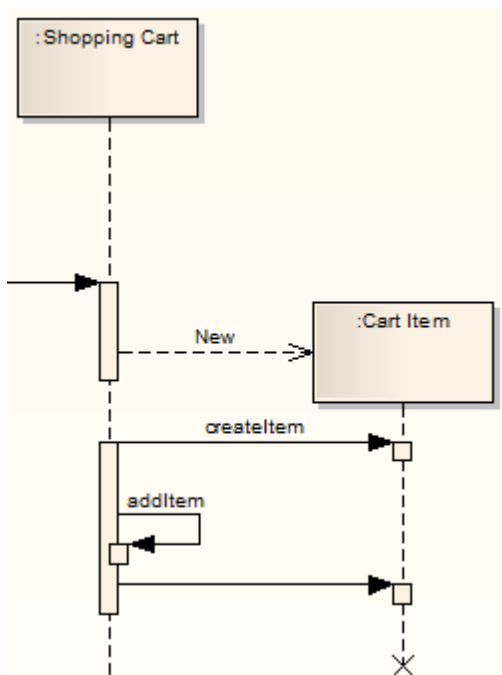
6.3.6.2 Denote Lifecycle of an Element

To capture element lifetimes using messages that are denoted as New or Delete message types, follow the steps below

Step	Action
------	--------

1	Double-click on a message within a Sequence diagram to display the Message Properties dialog.
2	In the Lifecycle field, click on the drop-down arrow and select New or Delete .
3	Click on the OK button to save the changes.

Example Diagram The example below shows two elements that have specific creation and deletion times



Notes

- To show the termination **X** on the lifeline in the example diagram, you must switch on garbage collection: **Tools | Options | Diagram > Sequence: Garbage Collect**

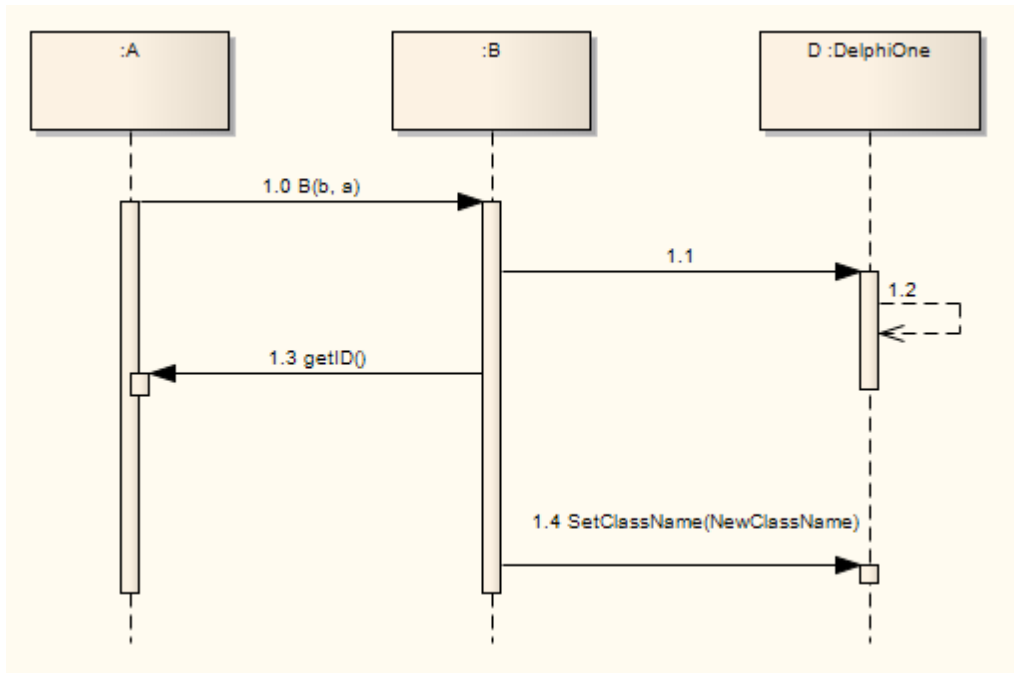
6.3.6.3 Layout of Sequence Diagrams

To offset the vertical height of sequence messages to get an attractive and effective layout

Step	Action
1	Select the appropriate message in a Sequence diagram.
2	Use the mouse to drag the message up or down as required. As you drag a message up or down a lifeline, any messages or fragments below that message are shifted up or down the same amount; however, be

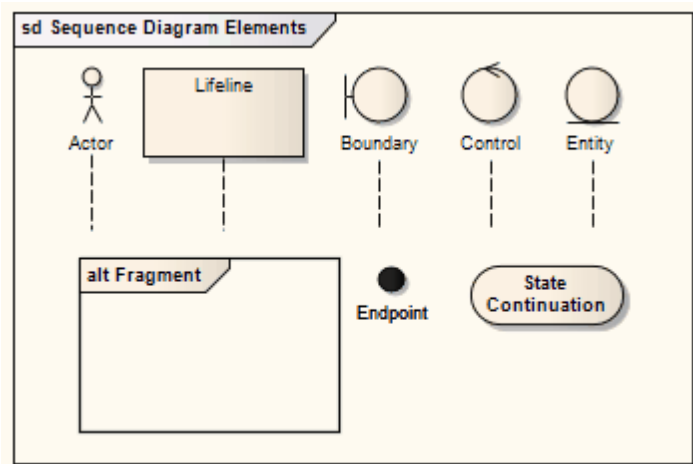
aware that if you drag up or down past the next or previous message, Enterprise Architect interprets that as the requirement to swap positions, rather than simply offset a message position.

Example Diagram The example below shows an economical use of space in a Sequence diagram.



6.3.6.4 Sequence Elements

The example below shows some possible elements of Sequence diagrams and their stereotyped display.



Element	Description	See also
Actor	An instance of an actor at runtime; this can be depicted either as the human figure or in rectangle notation.	Actor ^[1107]
Lifeline	An Object element with the stereotype Lifeline.	Lifeline ^[1144]
Boundary	Represents a user interface screen or input/output device.	Boundary ^[1663]
Entity	A persistent element - typically implemented as a database table or element.	Control ^[1665]
Control	The active component that controls what work gets done, when and how.	Entity ^[1666]

6.3.6.5 Sequence Diagrams and Version Control

You might create Sequence diagrams that use elements from other packages as the Lifelines within the diagram. In such cases, the diagrams could be corrupted when the element packages are checked in and out under version control. This is because during checkout the elements are first deleted from the model and then re-imported, and although they are reinstated in the diagrams, any Messages connecting them are not.

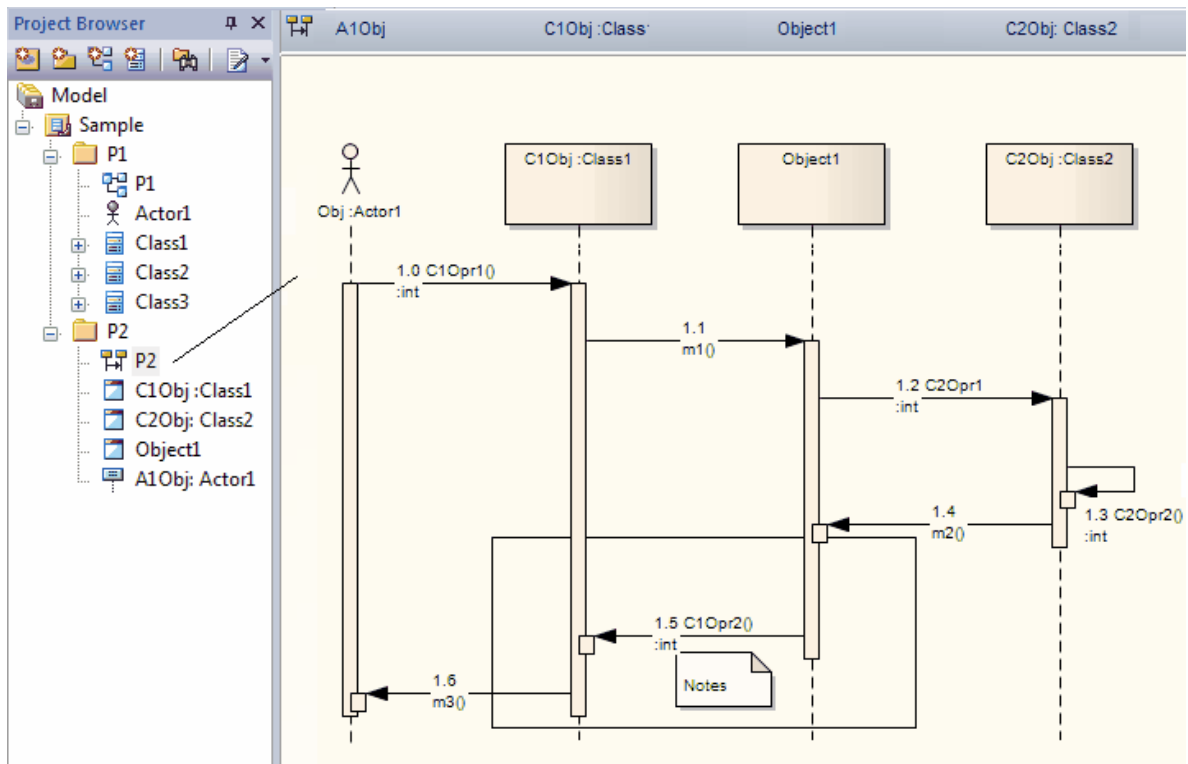
So, if the diagram and its elements reside in different packages, a round-trip of the element package through version control might damage the Sequence diagram.

The solution is to drag-and-drop each Class onto the Sequence diagram as an *object* - when you drop the Class onto the Sequence diagram, in the Paste Element dialog select the **as Instance of Element (Object)** option. This creates a new object in the *diagram's* parent package, based on the selected Class element. You then create the Messages between the objects.

Therefore, to ensure that a Sequence diagram is not damaged by round-trips of other packages through version control, remember that:

- The Lifelines must be objects (even though Enterprise Architect allows you to drop elements as Lifelines onto a Sequence diagram, it is not a strictly UML compliant construct)
- The Lifelines must be in the same package as the diagram.

The following illustration shows the Project Browser with two packages: *P1*, containing the elements, and *P2*, containing a Sequence diagram that uses those elements. The diagram itself is also shown.



This diagram is not damaged, because all the Lifelines are objects and these objects reside in the same package as the Sequence diagram.

6.3.6.6 Sequence Element Activation

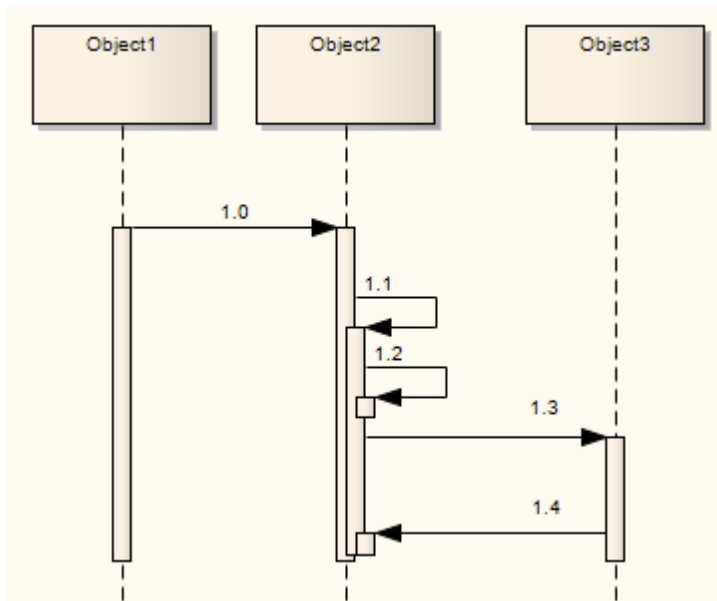
Sequence elements in a Sequence diagram have Activation rectangles drawn along their lifelines. These rectangles describe the time the element is active during the overall period of processing. This visual representation can be suppressed by right-clicking the Sequence diagram, and selecting the **Suppress Activations** context menu option.

In general, Enterprise Architect calculates the period of activation for you, but in some cases you might want to fine tune the rectangle length. There are several context menu options on a sequence message that you can use to accomplish this. To access the following context menu, right-click on the message and select the **Activations** context menu option.

Command	Description
Start New Message Group	Starts off a new round of processing in the current diagram. This enables you to describe more than one processing scenario in a single diagram.
Extend Source Activation Down	Forces an element to stay active beyond the normal processing period. This could be used to express an element that continues its own processing concurrently with other processes.

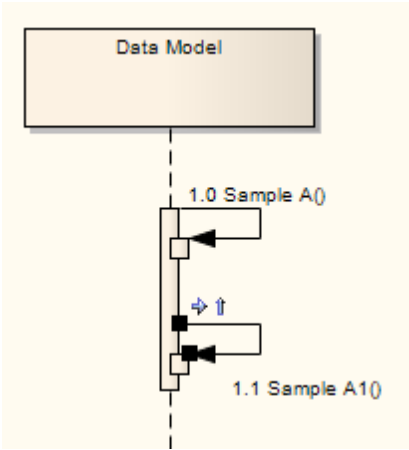
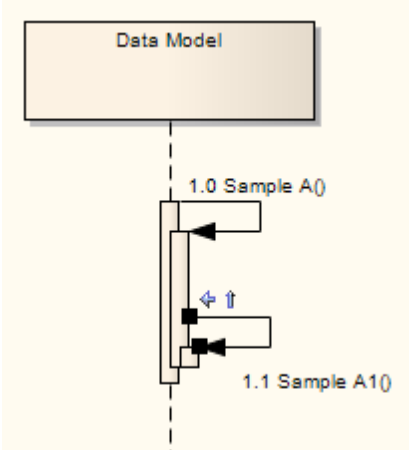
Extend Source Activation Up	Forces an element's activation upwards.
End Source Activation	Truncates the activation of the source element after the current message. This is useful for expressing an asynchronous message after which the source element becomes idle.
End Target Activation	Ends a Forced Activation started by the Extend Source Activation options.
Raise Activation Level	Displays on the context menu only where its use is appropriate. For example, after a self-message the next message starts by default at a lower activation level but the Raise Activation Level command displays on the context menu to enable you to raise its level.
Lower Activation Level	Displays on the context menu only where its use is appropriate.

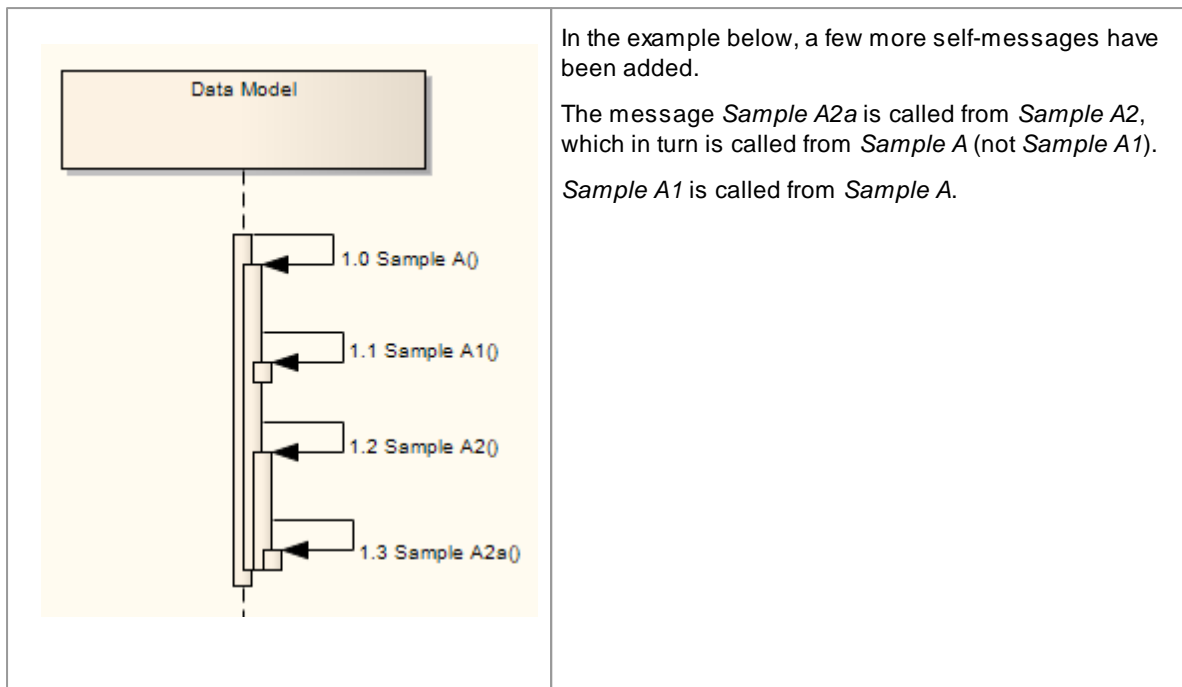
A more convenient way to change activation levels is directly on the diagram. Whenever appropriate, left and/or right arrows display on specific connectors. In the following diagram, see connector 1.3. Click on the arrow to raise or lower the activation level.



6.3.6.7 Lifeline Activation Levels

Complicated processing systems can be easily negotiated and reflected in Sequence diagrams, by adding activation layers on a single lifeline.

Example	Description
	<p>A Class invokes the method <i>Sample A</i>, which in turn calls <i>Sample A1</i>.</p> <p>To produce the arrangement in the diagram:</p> <ol style="list-style-type: none"> 1. Select the More tools Interaction menu option. 2. Click on the Self-message icon in the Interaction Relationships panel. 3. Then click on the lifeline.
	<p>In order to raise the Activation level of <i>Sample A1</i>, click on the raise arrow of the selected connector.</p> <p>The lifeline now visually depicts that method <i>Sample A1</i> is called during the processing of <i>Sample A</i>.</p>



In the example below, a few more self-messages have been added.

The message *Sample A2a* is called from *Sample A2*, which in turn is called from *Sample A* (not *Sample A1*).

Sample A1 is called from *Sample A*.

6.3.6.8 Sequence Message Label Visibility

How to

To hide and show the labels used in Sequence messages

Step	Action
1	Right-click on the message within the Sequence diagram and select the Set Label Visibility context menu option. The Label Visibility dialog displays.
2	Select or clear the checkbox against each message label to display or hide, respectively.
3	Click on the OK button to save the settings.

6.3.6.9 Change the Top Margin

In order to change the top margin of a Sequence diagram from the default 50 units, right-click on the diagram and select the **Set Top Margin** context menu option. You can set the top margin to any value between 30 and 250 units.

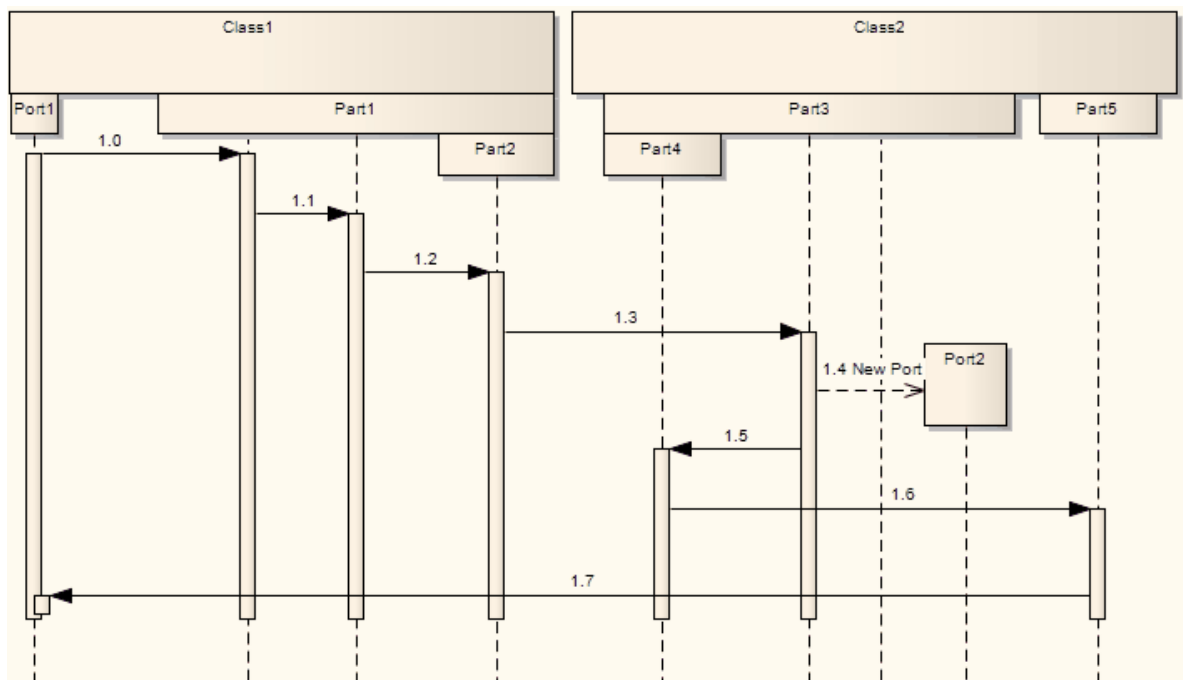
6.3.6.10 Inline Sequence Elements

It is possible to represent Part and Port elements on a Sequence diagram. Child Parts and Ports appear as inline sequence elements under their parent Class sequence element.

How to

To represent Part and Port elements on a Sequence diagram

Step	Action
1	Right-click on the sequence elements containing the child Ports or Parts, and select the Embedded Elements Embedded Elements context menu option.
2	Select the checkbox against each Part or Port to show, and click on the Close button.



6.3.7 Communication Diagram

A Communication diagram is a diagram that shows the interactions between elements at run-time in much the same manner as a Sequence diagram. However, Communication diagrams are used to visualize inter-object relationships, while Sequence diagrams are more effective at visualizing processing over time.

Communication diagrams employ ordered, labeled associations to illustrate processing. Numbering is important to indicate the order and nesting of processing. A numbering scheme could be:

- 1
- 1.1










1.1.1
1.1.2
1.2, and so on.

A new number segment begins for a new layer of processing, and would be equivalent to a method invocation.

Example Diagram [Example Communication Diagram](#) ^[1085]

Tools

Select Communication diagram elements and connectors from the Communication pages of the Toolbox. Click on the following elements and connectors for more information.

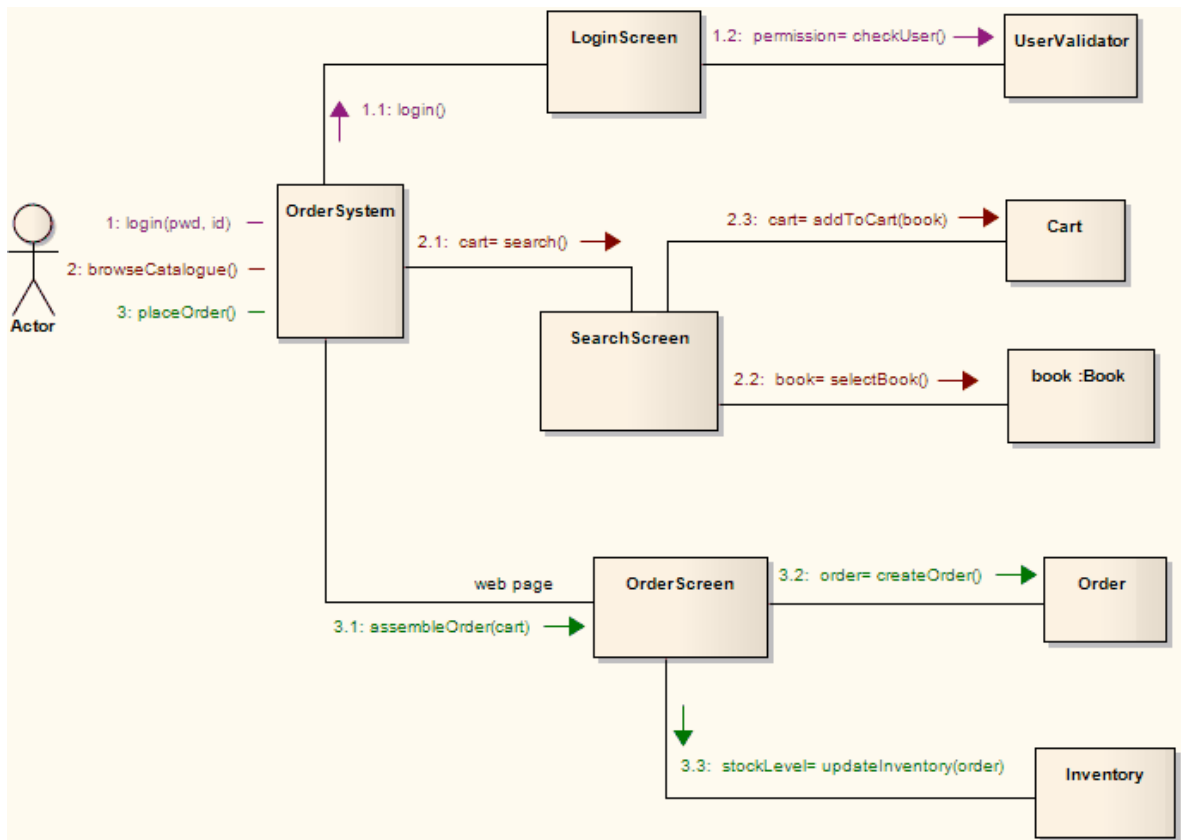
Communication Diagram Elements	Communication Diagram Connectors
 Actor	 Associate
 Object	 Nesting
 Boundary	 Realize
 Control	
 Entity	
 Package	

Learn more

- [Communication Diagrams in Color](#) ^[1086]

6.3.7.1 Example Communication Diagram

This example illustrates a Communication diagram among cooperating object instances. Note the use of message levels to capture related flows, and the different colors of the messages.



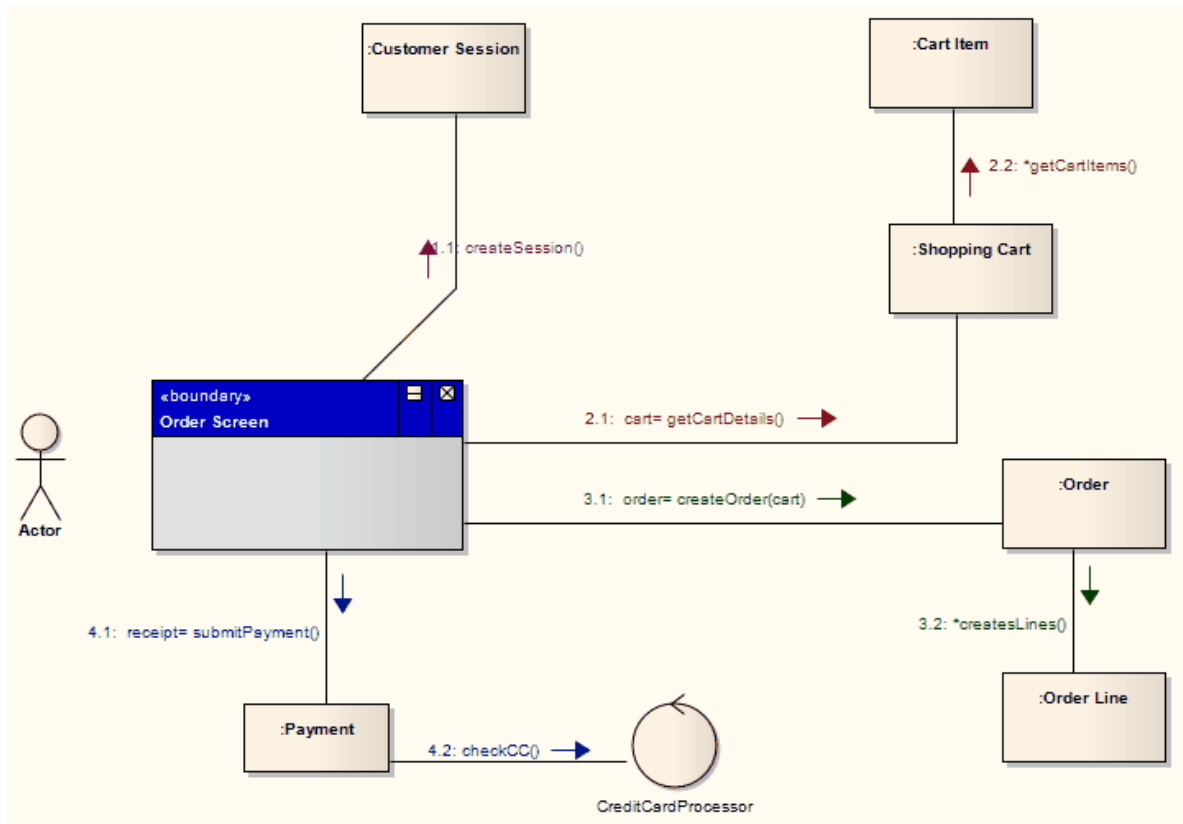
6.3.7.2 Communication Diagrams in Color

Enterprise Architect enables you to highlight particular message flows in a Communication diagram using different colors for each message set.

How to

To highlight the colors in a Communication diagram

Step	Action
1	Select the Tools Options Communication Colors menu option. The Communication Message Coloring page of the Options dialog displays.
2	Select the Use Communication Color checkbox.
3	Click on the drop-down arrow of each Message n field, and select the required color for each message group.
4	Click on the Close button. On your Communication diagram, each sequence group of messages displays in a different color, as shown below.



6.3.8 Interaction Overview Diagram

Interaction Overview diagrams visualize the cooperation between other interaction diagrams to illustrate a control flow serving an encompassing purpose. As Interaction Overview diagrams are a variant of Activity diagrams, most of the diagram notation is the same, as is the process of constructing the diagram.

Decision points, Forks, Joins, Start points and End points are the same. Instead of Activity elements, however, rectangular elements are used. There are two types of these elements:

















- Interaction elements display an inline Interaction diagram, which can be any one of the four types (Sequence, Timing, Communication or Interaction Overview)
- Interaction Occurrence elements are references to an existing Interaction diagram: they are visually represented by a frame, with **ref** in the frame's title space; the diagram name is indicated in the frame contents

To create an Interaction Occurrence, simply drag an Interaction diagram from the Project Browser onto your Interaction Overview diagram. The **ref** frame displays, encapsulating an instance of the Interaction diagram.

Example Diagram [Example Interaction Overview Diagram](#) ¹⁰⁸⁸

Tools

Select Interaction Overview diagram elements and connectors from the Activity pages of the Toolbox. Click on the following elements and connectors for more information.

Interaction Overview Diagram Elements	Interaction Overview Diagram Connectors
 Partition	 Control Flow
 Decision	 Object Flow
 Send	 Interrupt Flow
 Receive	
 Synch	
 Initial	
 Final	
 Flow Final	
 Region	
 Exception	
 Merge	
 Fork/Join	
 Fork/Join	

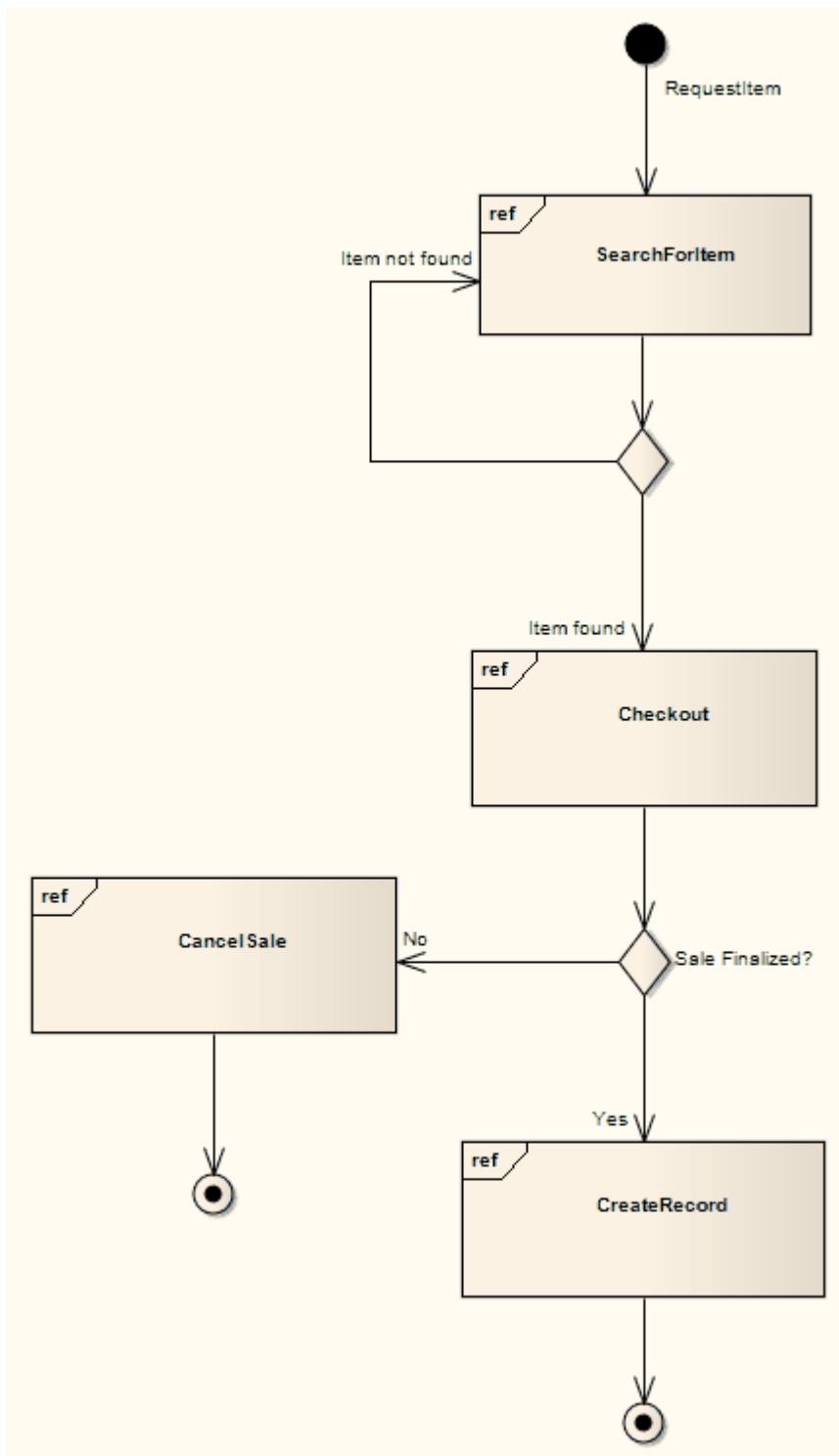
Learn more

- [Activity diagrams](#) ^[1028]
- [Interaction](#) ^[1137]
- [Interaction Occurrence](#) ^[1138]

6.3.8.1 Example Interaction Overview Diagram

This diagram depicts a sample sale process, shown in an Interaction Overview diagram, with sub-processes abstracted within Interaction Occurrences.

The diagram appears very similar to an Activity diagram, and is conceptualized the same way; as the flow moves into an interaction, the respective interaction's process must be followed before the Interaction Overview's flow can advance.



6.4 UML Elements

Models in UML are constructed from elements such as *Classes*, *Objects*, *Interfaces*, *Use Cases*, *Components* and *Nodes*, each of which has a different purpose, different rules and different notation. Model elements are used at different stages of the design process for different purposes. For example,

- During early analysis, Use Cases, Activities, Business Processes, Objects and Collaborations are used to capture the problem domain
- During elaboration, Sequence diagrams, Objects, Classes and State Machines are used to refine the system specification
- Components and Nodes are used to model larger parts of the system as well as the physical entities that are created and deployed into a production environment

Topics

Topic	Detail	See also
Behavioral Diagram Elements	Behavioral diagrams depict the behavioral features of a system or business process. Elements that can appear on Behavioral diagrams include Activity, Interaction, Lifeline, State Machine and Use Case.	Behavioral Diagram Elements ^[1090]
Structural Diagram Elements	Structural diagrams depict the structural elements composing a system or function. Elements that can appear on Structural diagrams include Class, Component, Interface, Node and Package.	Structural Diagram Elements ^[1179]

6.4.1 Behavioral Diagram Elements

The following elements are used in UML Behavioral Diagrams. For more information on using each element, click on the element name in this list:

- [Action](#) ^[1091], [Activity](#) ^[1101], [Actor](#) ^[1107]
- [Central Buffer Node](#) ^[1107], [Choice](#) ^[1108], [Collaboration](#) ^[1184], [Collaboration Use](#) ^[1186], [Combined Fragment](#) ^[1110]
- [Datastore](#) ^[1116], [Decision](#) ^[1117], [Diagram Frame](#) ^[1119], [Diagram Gate](#) ^[1120]
- [Endpoint](#) ^[1121], [Entry Point](#) ^[1123], [Exception](#) ^[1123], [Expansion Region](#) ^[1124], [Exit Point](#) ^[1126]
- [Final](#) ^[1127], [Flow Final](#) ^[1128], [Fork](#) ^[1130]
- [History](#) ^[1134]
- [Initial](#) ^[1136], [Interaction](#) ^[1137], [Interaction Occurrence](#) ^[1139], [Interruptible Activity Region](#) ^[1141]
- [Join](#) ^[1130], [Junction](#) ^[1142]
- [Lifeline](#) ^[1144]
- [Merge](#) ^[1145], [Message Endpoint](#) ^[1145], [Message Label](#) ^[1146]
- [Note](#) ^[1147]
- [Object](#) ^[1196]

- [Package](#)^[1200], [Partition](#)^[1148]
- [Receive](#)^[1149], [Region](#)^[1150]
- [Send](#)^[1151], [State](#)^[1152], [State/Continuation](#)^[1153], [State Lifeline](#)^[1158], [State Machine](#)^[1161], [Structured Activity](#)^[1161],
[Synch](#)^[1169], [System Boundary](#)^[1170]
- [Terminate](#)^[1172], [Trigger](#)^[1173]
- [Use Case](#)^[1174]
- [Value Lifeline](#)^[1177]

Learn more

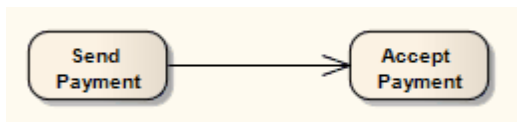
- [Behavioral Diagrams](#)^[1027]

6.4.1.1 Action



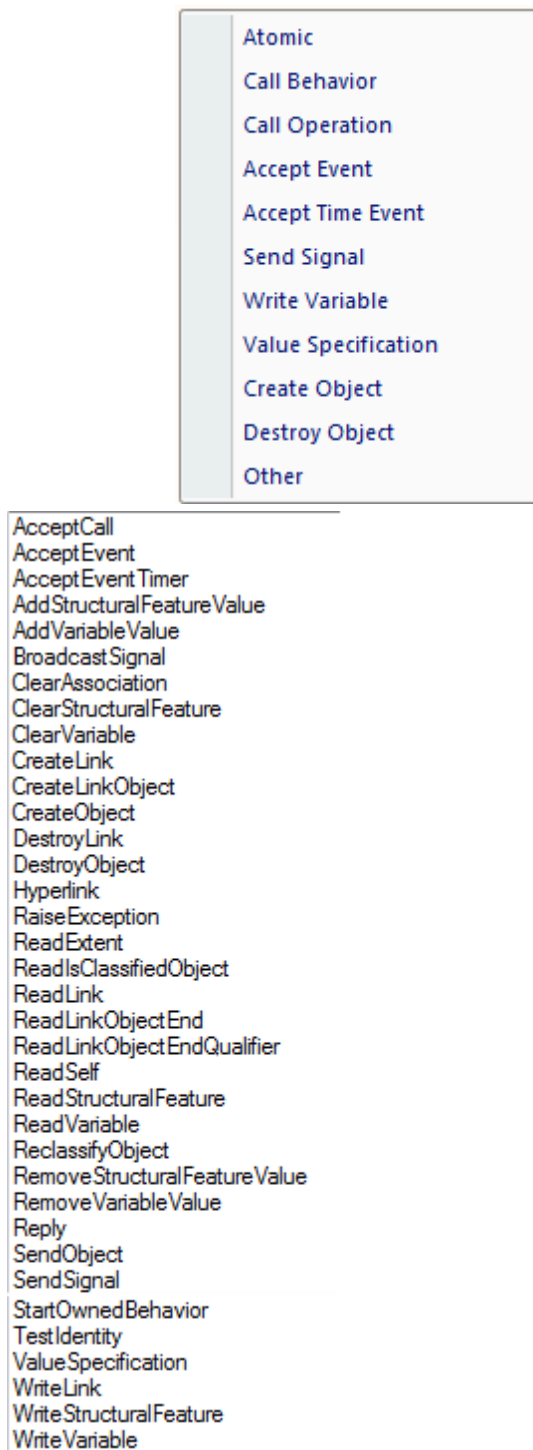
Description

An Action element describes a basic process or transformation that occurs within a system. It is the basic functional unit within an Activity diagram. Actions can be thought of as children of Activities. Both represent processes, but Activities can contain multiple steps or decomposable processes, each of which can be embodied in an Action. An Action cannot be further broken down or decomposed.



For a basic (Atomic) Action, you can define the effect of the Action using the Effect page of the element Properties dialog, and select to display the effect on the diagram. An Action can be further defined with pre-condition and post-condition notes.

Certain properties can be graphically depicted on the Action. When you first drag the Action icon from the Toolbox onto a diagram, Enterprise Architect prompts you to select from a list of the more common types of Action to create. If you select the **Other** option on this list, the New Action dialog displays; the **Other** drop-down list on this dialog enables you to select a more specialized type of Action from a complete list of Action types.



If you later decide that the Action type is not appropriate, you can change it on the Advanced page of the Action element Properties dialog - select the required new type from the **Kind** drop-down list. For a Value Specification Action, you can also set the value on this page.

The data values passed out of and into an Action can be represented by Action Pins. For an Action type other than a basic Action, you can also assign Action Pins to represent specific properties.

An Action can also be depicted as an Expansion Node to indicate that the Action comprises an Expansion Region.

Toolbox icon



Learn more

- [Activity Element](#)^[1107]
- [Activity Diagram](#)^[1028]
- [Local Pre/Post Conditions](#)^[1093]
- [Action Notation](#)^[1095]
- [Action Pin](#)^[1097]
- [Assign Action Pins](#)^[1098]
- [Action Expansion Node](#)^[1099]
- [Expansion Region](#)^[1124]
- [Class Operations in Activity Diagrams](#)^[1100]
- [Code Generation - Activity Diagrams](#)^[1787]
- [Add Action As Hyperlink](#)^[1670]

OMG UML Specification:

The OMG UML specification (UML Superstructure Specification, v2.1.1, p. 241) states:

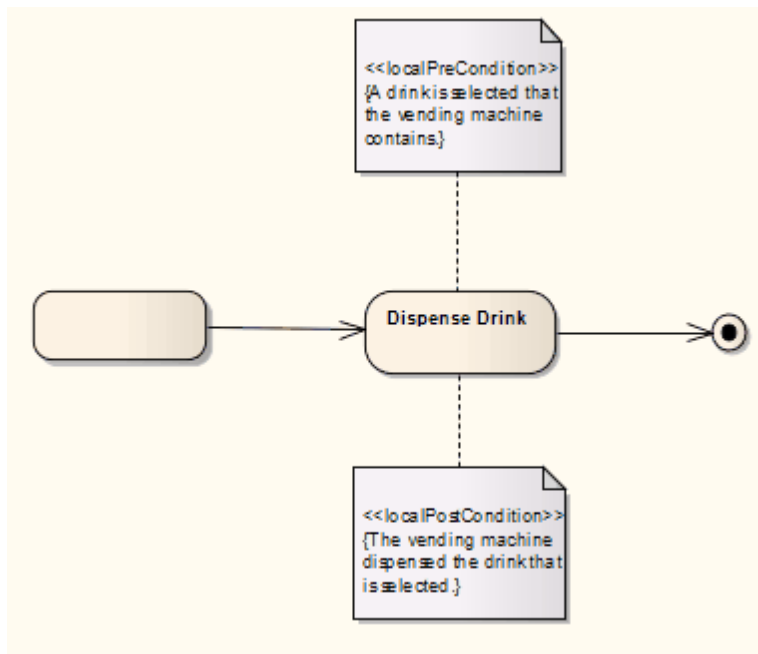
An action is a named element that is the fundamental unit of executable functionality. The execution of an action represents some transformation or processing in the modeled system, be it a computer system or otherwise.

The OMG UML specification (UML Superstructure Specification, v2.1.1, p. 313) also states:

An action may have sets of incoming and outgoing activity edges that specify control flow and data flow from and to other nodes. An action will not begin execution until all of its input conditions are satisfied. The completion of the execution of an action may enable the execution of a set of successor nodes and actions that take their inputs from the outputs of the action.

6.4.1.1.1 Local Pre/Post Conditions

Actions can be further defined with pre-condition and post-condition notes, which constrain an Action's entry and exit. These notes can be added to an Action as defined below.



How to

To attach a constraint to an Action

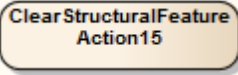
Step	Action
1	Right-click on the Action and select the Add Constraint context menu option. A <i>Note</i> is created on the diagram, connected to the Action.
2	Right-click on the <i>Note</i> and select the View Properties context menu option. The Constraint dialog displays.
3	In the Constraint Type field, click on the drop-down arrow and select the required constraint type.
4	In the Constraint field, type the text for the constraint.
5	Click on the OK button to save the constraint.

Learn more



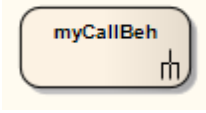
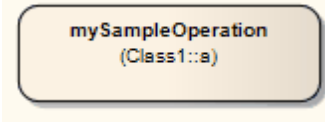

- [Action](#)^[1094]

6.4.1.1.2 Action Notation

Actions of most types are depicted as a round-cornered rectangle containing the Action type and Action name, as follows:



Other types have their own specific notation, as described below:

Action Kind	Notation	See also
AcceptEvent		AcceptEvent Actions ^[1095]
AcceptEventTimer		
CallBehavior		
CallOperation		
SendSignal		SendSignal Action & BroadcastSignal Action ^[1096]

AcceptEvent Actions:

For an *Accept Event* Action element, the Properties dialog contains a Triggers tab on which you define one or more triggers to denote the type of events accepted by the Action, as defined in the following table:

Field	Usage	See also
Name	Specify the name of the trigger.	

Field	Usage	See also
Type	<p>Specify the type of trigger: Call, Change, Signal or Time:</p> <ul style="list-style-type: none"> • Call - specifies that the event is a CallEvent, which sends a message to the associated object by invoking an operation • Change - specifies that the event is a ChangeEvent, which indicates that the transition is the result of a change in value of an attribute • Signal - specifies that the event is a SignalEvent, which corresponds to the receipt of an asynchronous signal instance • Time - corresponds to a TimeEvent; which specifies a moment in time <p>Code generation for State Machines currently supports Change and Time trigger events only, and expects a specification value.</p>	
Specification	Specify the event instigating the Transition.	

SendSignal Action & BroadcastSignal Action:

For a *SendSignal* or *BroadcastSignal* Action element, you can model the signal to be sent and the associated arguments to be conveyed, using the Signal tab of the element Properties dialog.

To model the signal to be sent and the associated arguments to be conveyed

Step	Action
1	In the Signal field, click on the (...) button and select the required signal from the Select Signal dialog.
2	In the Attribute field, click on the drop-down arrow and select the attribute (as previously created in the Signal element) with which the arguments are to be associated.
3	In the Value field, type the appropriate value for the attribute.
4	Identify the arguments (as ActionPins) for the Signal; click on the Add button under the Arguments panel, and select the appropriate Pins from the Select Pin dialog. To assign more than one Pin, press (Ctrl) whilst you select each Pin.

Step	Action
5	Click on the Save button.

Structural Feature Actions:

Enterprise Architect supports the following types of *Structural Feature* Actions:

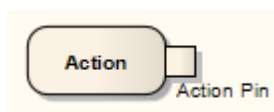
- AddStructuralFeatureValue
- ClearStructuralFeature
- ReadStructuralFeature
- RemoveStructuralFeatureValue
- WriteStructuralFeature

These actions can take Ports, Parts or Attributes as the target structural feature.

To set the appropriate target structure feature

Step	Action
1	Right-click on the Action element in the diagram, and select the Advanced Set Structural Feature context menu option. The Set Structural Feature dialog displays.
2	To locate the structural feature, click on the Add button. The Select Property dialog displays (a variant of the Select <Item> dialog).
3	Browse or search for the appropriate structural feature, and double-click on it. The feature name and location displays in the structuralFeature field of the Set Structural Feature dialog.
4	Click on the OK button to save the setting.

6.4.1.1.3 Action Pin

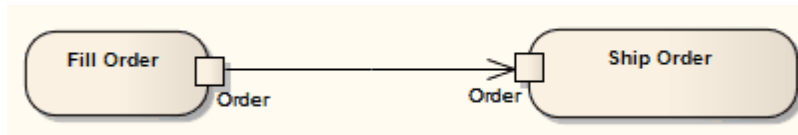


Description

An *Action Pin* is used to define the data values passed out of and into an Action. An *input pin* provides values

to the Action, whereas an *output pin* contains the results from that Action.

Action Pins are used below to connect two Actions:



Action Pins can be further characterized as defining exception parameters, streams, or states. Associating a state with a Pin defines the state of input or output values. For instance, the Pin could be called *Orders*, but the state could be *Validated* or *Canceled*.

To *add* an Action Pin to an Action, right-click on the Action to display the context menu and select the **Embedded Elements | Add Action Pin** menu option. (You can also assign Action Pins, to define specific properties of the Action.)

The Properties dialog of an Action Pin has a Pin tab on which you define the specific actions of the Pin.

A Pin serves as an argument for Call Behavior Actions and Call Operation Actions. When an Action is associated with a valid behavior in the model, the associated behavior's parameters are listed in the **Parameter** field drop-down list to facilitate a one-to-one mapping between the argument and the parameter. The fields in the Argument panel are enabled only for Pins belonging to Call Actions, and only when the Action is associated with a valid behavior with valid parameters.

You can also change certain properties of an Action Pin on the Advanced page of the element Properties dialog: right-click on the Pin and select the **Properties** context menu option. The *objectState* and *kind* properties can be set.

Learn more

- [Action](#)^[1097]
- [Assign Action Pins](#)^[1098]

6.4.1.1.4 Assign Action Pins

Apart from *adding* Action Pins to any Action, you can *assign* specialized input or output Action Pins to Actions that have a specific type (that is, those that are not Basic or Atomic Actions). These input/output Pins signify various *properties* of the Action - they are not visible as structures on the diagram unless they have previously been added, but are listed in the Project Browser as properties of the Action.

You can only assign Pins that have already been added or assigned to the Action, or that are being created specifically to be assigned to the Action.

How to

To assign Pins to an Action

Step	Action	See also
1	Right-click on the Action in the diagram, and select the Advanced Assign	

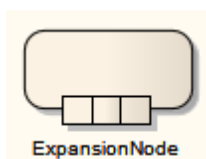
Step	Action	See also
	<p>Action Pins context menu option.</p> <p>The Assign Action Pins to <ActionName> dialog displays.</p> <p>The format of this dialog depends on the type of Action; for a:</p> <ul style="list-style-type: none"> • <i>SendObject</i> Action the dialog has two fields (request and target) • <i>TestIdentity</i> Action, three • <i>CallBehavior</i> Action, one (result) <p>The fields are populated in exactly the same way.</p>	
2	<p>The mandatory number and type of Pins are automatically selected (if they exist) or created.</p> <p>To change or add a Pin in a field, click on the corresponding Add button.</p> <p>The Select Pins dialog displays, showing the selected Action and listing all the input Pins currently owned by the Action.</p>	Select <Item> Dialog ¹⁸⁸⁰
3	<p>Double-click on one of the Pins (or, depending on the multiplicity of the Pin, (Ctrl) +click on several Pins).</p> <p>Alternatively, if no suitable Pin exists, click on the Add New button and then click on the newly-created Pin.</p> <p>The selected Pin is identified in the field on the Assign Action Pins to <ActionName> dialog.</p>	
4	Click on the OK button.	

To check the exact location of an assigned Action Pin, you can right-click on the Pin name in the dialog and select the **Find in Project Browser** context menu option.

Learn more

- [Action Pin](#) ¹⁰⁹⁷

6.4.1.1.5 Action Expansion Node



Description

Representing an Action as an *Expansion Node* is a shorthand notation to indicate that the Action comprises an Expansion Region.

To specify an Action as an Expansion Node, right-click on the Action and select the **Embedded Elements | Add Expansion Node** context menu option.

After designating an Action as an Expansion Node, you can modify or delete it using the **Embedded Elements | Embedded Elements** menu option.

Learn more

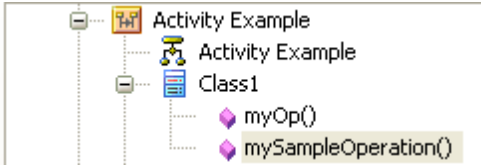
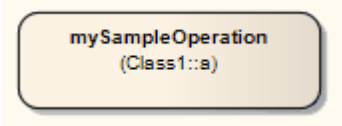
- [Action](#)^[1097]
- [Expansion Region](#)^[1124]

6.4.1.1.6 Class Operations in Activity Diagrams

Operations from Classes can be displayed on Activity diagrams as Actions. When an operation is shown as an Action, the notation of the Action displays the name of the Class that features the operation.

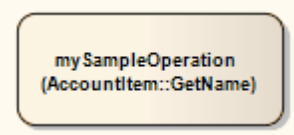
How to

To add an operation to an Activity diagram

Step	Action	See also
1	Open an Activity diagram.	
2	From the Project Browser open a Class and locate the operation to be added to the Activity diagram.	
3	Drag the operation on to the diagram. 	
4	When the operation has been added to the Activity diagram, the Action displays the name of the Class that features the operation.  Right-click on the Action in the diagram to locate, in the Project Browser, the: <ul style="list-style-type: none"> • Behavior classifier (CallBehavior Activity) using the Find Locate 	

Step	Action	See also
	<p>Classifier in Project Browser context menu option or</p> <ul style="list-style-type: none"> Call operation (CallOperation Activity) using the Find Locate Operation in Project Browser context menu option 	

If it becomes necessary to change the operation that this Action refers to

Step	Action	See also
1	<p>Right-click on the Action and select the Advanced Set Operation menu option.</p> <p>The Set Operation dialog displays.</p>	Set Operation dialog ^[884]
2	<p>If necessary, in the In Namespace field, select the model that contains the required operation.</p>	
3	<p>Double-click on the required operation.</p> <p>The Action updates to show the new classifier and operation.</p> 	

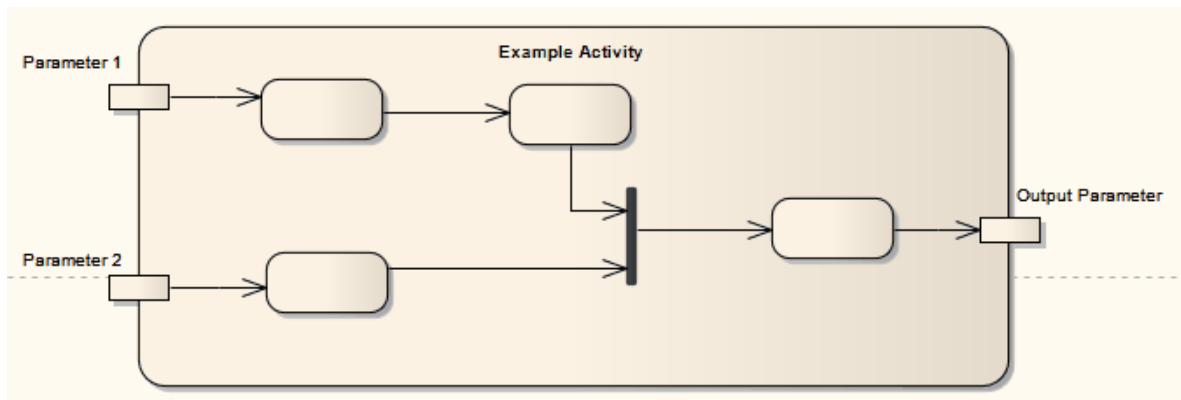
6.4.1.2 Activity



Description

An Activity organizes and specifies the participation of subordinate behaviors, such as sub-Activities or Actions, to reflect the control and data flow of a process. Activities are used in Activity diagrams for various modeling purposes, from procedural-type application development for system design, to business process modeling of organizational structures or work flow.

The following simple diagram of an Activity contains Action elements and includes input parameters and output parameters.



You can define an Activity as a composite element, either during creation or during later edits. When creating a composite Activity element, it is simpler to apply the mechanism for creating Structured Activity elements, which reduces the number of steps to work through. See the *Structured Activity* topic. If converting an existing Activity element, right-click on the element and select the **Advanced | Make Composite** context menu option.

Certain properties can be graphically depicted on an Activity. The Actions in an Activity can be further organized by Activity Partitions.

Toolbox icon



Learn more

- [Activity Notation](#) ^[1103]
- [Activity Parameter Nodes](#) ^[1103]
- [Activity Partition](#) ^[1105]
- [Action](#) ^[1091]
- [Activity Diagram](#) ^[1028]
- [Composite Element](#) ^[825]
- [Structured Activity](#) ^[1161]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 318*) states:

An activity specifies the coordination of executions of subordinate behaviors, using a control and data flow model. The subordinate behaviors coordinated by these models may be initiated because other behaviors in the model finish executing, because objects and data become available, or because events occur external to the flow. The flow of execution is modeled as activity nodes connected by activity edges. A node can be the execution of a subordinate behavior, such as an arithmetic computation, a call to an operation, or manipulation of object contents. Activity nodes also include flow-of-control constructs, such as synchronization, decision, and concurrency control. Activities may form invocation hierarchies invoking other activities, ultimately resolving to individual actions. In an object-oriented model, activities are usually invoked indirectly as methods bound to operations that are directly invoked.

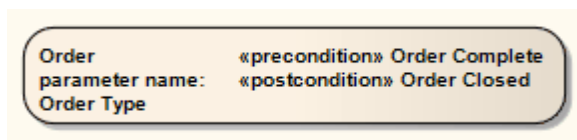
Activities may describe procedural computation. In this context, they are the methods corresponding to operations on classes. Activities may be applied to organizational modeling for business process engineering and workflow. In this context, events often originate from inside the system, such as the finishing of a task, but also from outside the system, such as a customer call. Activities can also be used for information system modeling to specify system level processes. Activities may contain actions of various kinds:

- Occurrences of primitive functions, such as arithmetic functions.
- Invocations of behavior, such as activities.
- Communication actions, such as sending of signals.
- Manipulations of objects, such as reading or writing attributes or associations.

Actions have no further decomposition in the activity containing them. However, the execution of a single action may induce the execution of many other actions. For example, a call action invokes an operation that is implemented by an activity containing actions that execute before the call action completes.

6.4.1.2.1 Activity Notation

Certain properties can be graphically depicted on an Activity element, as shown below.



To define these properties, right-click on the Activity and select the **Properties** context menu option, then select the Advanced page of the Properties dialog.

Learn more

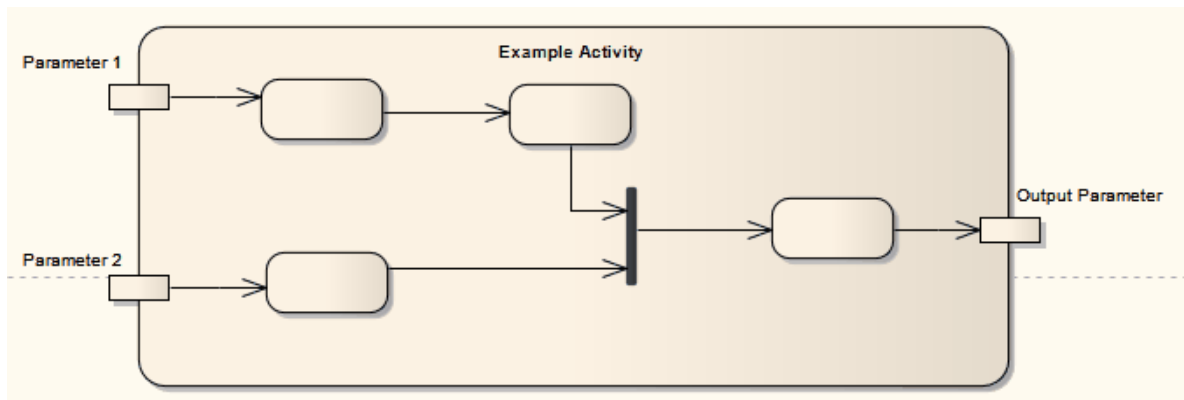
- [Activity](#)^[1103]

6.4.1.2.2 Activity Parameter Nodes

Description

An Activity Parameter Node accepts input to an Activity or provides output from an Activity.

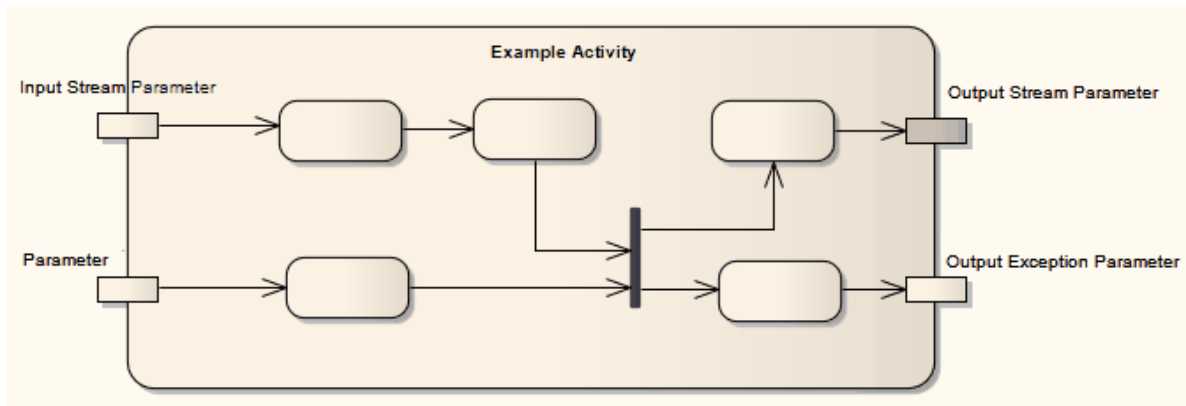
The following example depicts two entry parameters and one output parameter defined for the Activity.



How to

To define an Activity Parameter Node for an Activity

Step	Action
1	Right-click on the element and select the Embedded Elements Add Activity Parameter context menu option.
2	The Properties dialog displays, which prompts for the Name and other properties of the embedded element.
3	To further define the new Activity Parameter, select the Parameter page of the Properties dialog and provide: <ul style="list-style-type: none"> • Type • Default Value • Direction • Whether this is a fixed value • Multiplicity upper and lower bounds • Whether to allow duplicates and • Whether multiplicity is ordered <p>Activity Parameter Nodes also have the Exception and Stream options:</p> <ul style="list-style-type: none"> • Exception indicates that a parameter can emit a value at the exclusion of other outputs, usually because of some error • Stream indicates whether or not a parameter can accept or post values during the execution of the Activity



Learn more

- [Activity Element](#)^[1107]
- [Action Pin](#)^[1097]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 338*) states:

An activity parameter node is an object node for inputs and outputs to activities.

... Activity parameter nodes are object nodes at the beginning and end of flows that provide a means to accept inputs to an activity and provide outputs from the activity, through the activity parameters.

Activity parameters inherit support for streaming and exceptions from Parameter.

6.4.1.2.3 Activity Partition

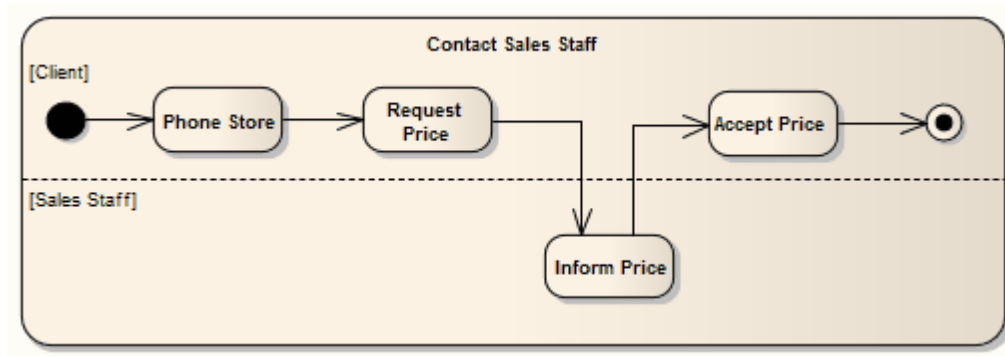
Description

Enterprise Architect supports two types of *Activity Partition*:

- The *Activity Partition feature*, described in this topic, which is used to logically organize an *Activity element*
- The *Activity Partition element*, which is used to logically organize an *Activity diagram*

In effect, these are the same. They partition the Actions of the Activity without affecting the token flow, helping to structure the view or parts of the Activity.

An example of a feature-partitioned Activity is shown below:



How to

To define Partitions using the feature

Step	Action
1	Right-click on the Activity element and select the Advanced Partition Activity context menu option. The Activity Partitions dialog displays.
2	In the Name field, type the name of a partition. Click on the Save button.
3	Repeat step 2 for each partition to be created.

Learn more

- [Activity Diagram](#) ^[1028]
- [Activity Element](#) ^[1107]
- [Activity Partition Element](#) ^[1148]

OMG UML Specification:

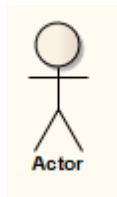
The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 341*) states:

Partitions divide the nodes and edges to constrain and show a view of the contained nodes. Partitions can share contents. They often correspond to organizational units in a business model. They may be used to allocate characteristics or resources among the nodes of an activity.

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 341*) also states:

An activity partition is a kind of activity group for identifying actions that have some characteristic in common.

6.4.1.3 Actor



Description

An Actor is a user of the system; user can mean a human user, a machine, or even another system or subsystem in the model. Anything that interacts with the system from the outside or system boundary is termed an Actor. Actors are typically associated with Use Cases.

Actors can use the system through a graphical user interface, through a batch interface or through some other media. An Actor's interaction with a Use Case is documented in a Use Case scenario, which details the functions a system must provide to satisfy the user requirements.

Actors also represent the role of a user in Sequence Diagrams, where you can display them using rectangle notation. Enterprise Architect supports a stereotyped Actor element for business modeling. The business modeling elements also represent Actors as stereotyped Objects.

Toolbox icon



Learn more

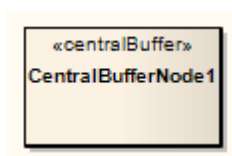
- [Sequence Diagrams](#)^[1074]
- [Business Modeling](#)^[1514]
- [Use Case](#)^[1174]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 584*) states:

An actor models a type of role played by an entity that interacts with the subject (e.g. by exchanging signals and data), but which is external to the subject. ... Actors may represent roles played by human users, external hardware, or other subjects. Note that an actor does not necessarily represent a specific physical entity but merely a particular facet (i.e., "role") of some entity that is relevant to the specification of its associated Use Cases. Thus, a single physical instance may play the role of several different actors and, conversely, a given actor may be played by multiple different instances.

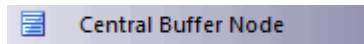
6.4.1.4 Central Buffer Node



Description

A Central Buffer Node is an object node for managing flows from multiple sources and destinations, represented in an Activity diagram. It acts as a buffer for multiple in-flows and out-flows from other object nodes, but does not connect directly to Actions.

Toolbox icon



Learn more

- [Activity Diagram](#) [1028]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 352*) states:

A central buffer node is an object node for managing flows from multiple sources and destinations. ... A central buffer node accepts tokens from upstream object nodes and passes them along to downstream object nodes.

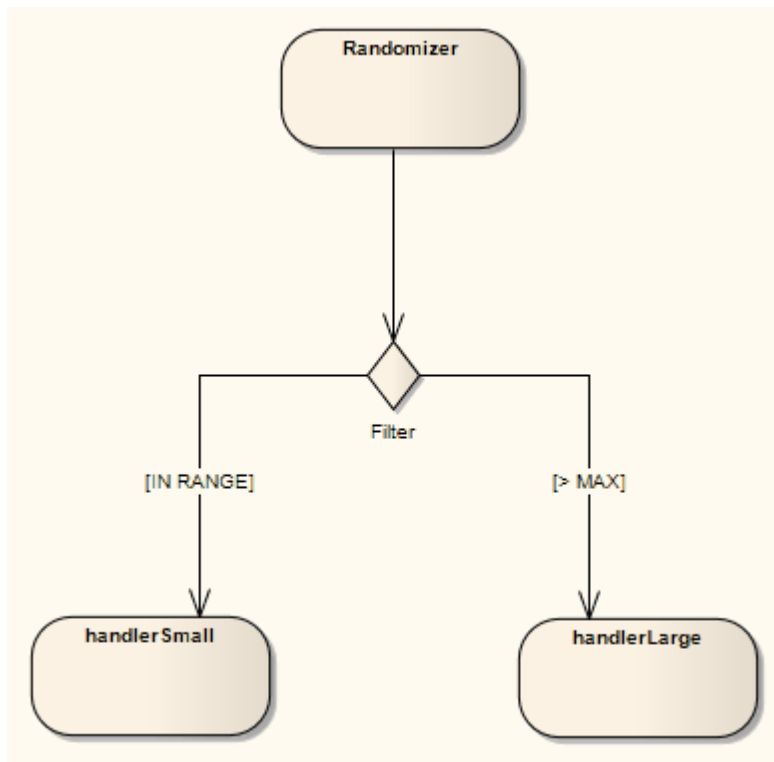
6.4.1.5 Choice



Description

The Choice pseudo-state is used to compose complex transitional paths in, for example, a State Machine diagram, where the outgoing transition path is decided by dynamic, run-time conditions. The run-time conditions are determined by the actions performed by the State Machine on the path leading to the choice.

The following example depicts the Choice element. Upon reaching the Filter pseudo-state, a transition fires to the appropriate state based on the run-time value passed to the Filter. Very similar in form to a Junction pseudo-state, the Choice pseudo-state's distinction is in deciding transition paths at run-time.



Toolbox icon



Learn more

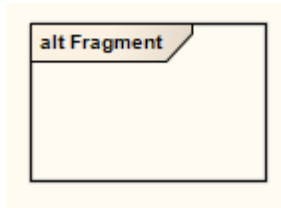
- [State Machine Diagram](#) ^[1032]
- [State Machine](#) ^[1152]
- [Pseudo-states](#) ^[1037]
- [Junction](#) ^[1142]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 538*) states:

...choice vertices which, when reached, result in the dynamic evaluation of the guards of the triggers of its outgoing transitions. This realizes a dynamic conditional branch. It enables splitting of transitions into multiple outgoing paths such that the decision on which path to take may be a function of the results of prior actions performed in the same run-to-completion step. If more than one of the guards evaluates to true, an arbitrary one is selected. If none of the guards evaluates to true, then the model is considered ill-formed. (To avoid this, it is recommended to define one outgoing transition with the predefined "else" guard for every choice vertex.) Choice vertices should be distinguished from static branch points that are based on junction points.

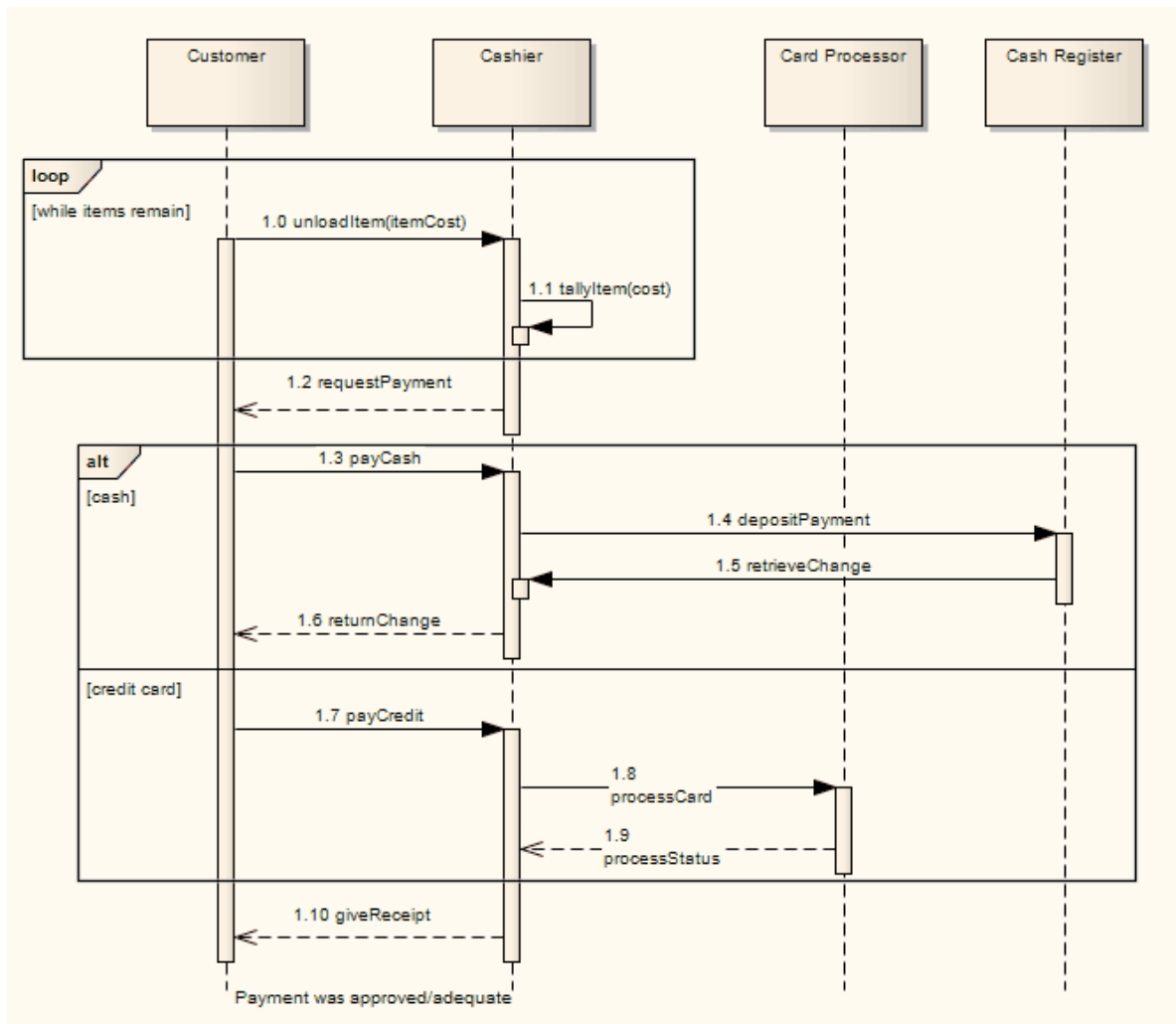
6.4.1.6 Combined Fragment



Description

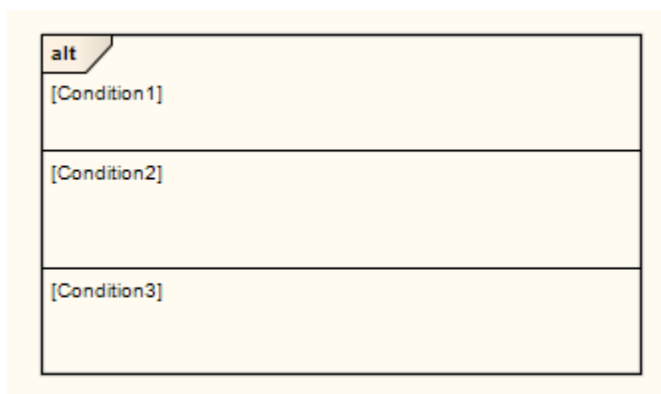
A Combined Fragment reflects a piece or pieces of interaction (called interaction operands) controlled by an interaction operator, whose corresponding boolean conditions are known as interaction constraints. It displays as a transparent window, divided by horizontal dashed lines for each operand.

The following diagram illustrates the use of Combined Fragments, with a Sequence diagram modeling a simplified purchasing process. A loop fragment is created to iterate through an unknown number of items for purchase, after which the cashier requests payment. At this point, two payment options are considered and an alternative fragment is created, divided to show the two operands: cash and credit card. After the fragment completes its trace, the cashier gives a receipt to the customer, under the fulfilled condition that payment requirements were met.



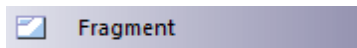
The order of interaction fragment conditions can be changed directly on the diagram:

1. Select an interaction fragment with more than one condition defined; up and down arrows appear on the right hand side of the each condition.
2. Click on the appropriate arrow to change the order.



Notes

- In order to select an interaction fragment, you must click near the inside edge or drag a selection rectangle around the fragment; this prevents accidental selection when moving connectors inside the interaction fragment
- Press and hold (**Alt**) to move a combined fragment independently of its contents

Toolbox icon**Learn more**

- [Sequence Diagram](#)^[1074]
- [Create a Combined Fragment](#)^[1112]
- [Interaction Operator](#)^[1113]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 467*) states:

A combined fragment defines an expression of interaction fragments. A combined fragment is defined by an interaction operator and corresponding interaction operands. Through the use of CombinedFragments the user will be able to describe a number of traces in a compact and concise manner.

6.4.1.6.1 Create a Combined Fragment**How to**

To create a Combined Fragment in Enterprise Architect

Step	Action	See also
1	Drag the <i>Fragment</i> element onto the diagram from the Interaction Elements page of the Toolbox.	Combined Fragment ^[1110]
2	In the Type field, click on the drop-down arrow and select one of the various types of interaction operator.	Interaction Operators ^[1113]
3	In the Condition field, specify a condition or interaction constraint for each operand.	
4	A rectangular frame displays, partitioned by dashed lines into segments for each operand.	

Step	Action	See also
5	Adjust the frame to encompass the required event occurrences for each operand.	

6.4.1.6.2 Interaction Operators

When creating Combined Fragments, you must apply an appropriate interaction operator to characterize the fragment. The following table provides guidance on the various operators, and their corresponding descriptions.

Interaction Operator	Action
alt	Divide up interaction fragments based on Boolean conditions.
opt	Enclose an optional fragment of interaction.
par	Indicate that operands operate in parallel.
loop	Indicate that the operand repeats a number of times, as specified by interaction constraints.
critical	Indicate a sequence that cannot be interrupted by other processing.
neg	Assert that a fragment is invalid, and implies that all other interaction is valid.
assert	Specify the only valid fragment to occur. This operator is often enclosed within a <i>consider</i> or <i>ignore</i> operand.
strict	Indicate that the behaviors of the operands must be processed in strict sequence.
seq	Indicate that the Combined Fragment is weakly sequenced. This means that the ordering within operands is maintained, but the ordering between operands is undefined, so long as an event occurrence of the first operand precedes that of the second operand, if the event occurrences are on the same lifeline.
ignore	Indicate which messages should be ignored during execution, or can appear anywhere in the execution trace.

Interaction Operator	Action
consider	Specify which messages should be considered in the trace. This is often used to specify the resulting event occurrences with the use of an assert operator.
ref	Provide a reference to another diagram. The ref fragment is not created using the method described in the <i>Create a Combined Fragment</i> topic. To create a ref fragment, simply drag an existing diagram from the Project Browser onto the current diagram.

Learn more

- [Combined Fragments](#)^[1110]
- [Create a Combined Fragment](#)^[1112]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 468-471*) states:

The semantics of a CombinedFragment is dependent upon the interactionOperator as explained below.

Alternatives

*The interactionOperator **alt** designates that the CombinedFragment represents a choice of behavior. At most one of the operands will be chosen. The chosen operand must have an explicit or implicit guard expression that evaluates to true at this point in the interaction. An implicit true guard is implied if the operand has no guard.*

The set of traces that defines a choice is the union of the (guarded) traces of the operands.

*An operand guarded by **else** designates a guard that is the negation of the disjunction of all other guards in the enclosing CombinedFragment.*

If none of the operands has a guard that evaluates to true, none of the operands are executed and the remainder of the enclosing InteractionFragment is executed.

Option

*The interactionOperator **opt** designates that the CombinedFragment represents a choice of behavior where either the (sole) operand happens or nothing happens. An option is semantically equivalent to an alternative CombinedFragment where there is one operand with non-empty content and the second operand is empty.*

Break

*The interactionOperator **break** designates that the CombinedFragment represents a breaking scenario in the sense that the operand is a scenario that is performed instead of the remainder of the enclosing InteractionFragment. A **break** operator with a guard is chosen when the guard is true and the rest of the enclosing Interaction Fragment is ignored. When the guard of the **break** operand is false, the **break** operand is ignored and the rest of the enclosing InteractionFragment is chosen. The choice between a **break** operand without a guard and the rest of the enclosing InteractionFragment is done non-deterministically.*

*A CombinedFragment with interactionOperator **break** should cover all Lifelines of the enclosing InteractionFragment.*

Parallel

The interactionOperator **par** designates that the CombinedFragment represents a parallel merge between the behaviors of the operands. The OccurrenceSpecifications of the different operands can be interleaved in any way as long as the ordering imposed by each operand as such is preserved.

A parallel merge defines a set of traces that describes all the ways that OccurrenceSpecifications of the operands may be interleaved without obstructing the order of the OccurrenceSpecifications within the operand.

Weak Sequencing

The interactionOperator **seq** designates that the CombinedFragment represents a weak sequencing between the behaviors of the operands.

Weak sequencing is defined by the set of traces with these properties:

1. The ordering of OccurrenceSpecifications within each of the operands is maintained in the result.
2. OccurrenceSpecifications on different lifelines from different operands may come in any order.
3. OccurrenceSpecifications on the same lifeline from different operands are ordered such that an OccurrenceSpecification of the first operand comes before that of the second operand.

Thus weak sequencing reduces to a parallel merge when the operands are on disjunct sets of participants. Weak sequencing reduces to strict sequencing when the operands work on only one participant.

Strict Sequencing

The interactionOperator **strict** designates that the CombinedFragment represents a strict sequencing between the behaviors of the operands. The semantics of strict sequencing defines a strict ordering of the operands on the first level within the CombinedIFragment with interactionOperator **strict**. Therefore OccurrenceSpecifications within contained CombinedFragment will not directly be compared with other OccurrenceSpecifications of the enclosing CombinedFragment.

Negative

The interactionOperator **neg** designates that the CombinedFragment represents traces that are defined to be invalid.

The set of traces that defined a CombinedFragment with interactionOperator negative is equal to the set of traces given by its (sole) operand, only that this set is a set of invalid rather than valid traces. All InteractionFragments that are different from Negative are considered positive meaning that they describe traces that are valid and should be possible.

Critical Region

The interactionOperator **critical** designates that the CombinedFragment represents a critical region. A critical region means that the traces of the region cannot be interleaved by other OccurrenceSpecifications (on those Lifelines covered by the region). This means that the region is treated atomically by the enclosing fragment when determining the set of valid traces. Even though enclosing CombinedFragments may imply that some OccurrenceSpecifications may interleave into the region, such as with **par**-operator, this is prevented by defining a region.

Thus the set of traces of enclosing constructs are restricted by critical regions.

Ignore / Consider

(p. 473) The interactionOperator **ignore** designates that there are some message types that are not shown within this combined fragment. These message types can be considered insignificant and are implicitly ignored if they appear in a corresponding execution. Alternatively one can understand **ignore** to mean that the messages that are ignored can appear anywhere in the traces.

Conversely the interactionOperator **consider** designates which messages should be considered within this CombinedFragment. This is equivalent to defining every other message to be ignored.

Assertion

The interactionOperator **assert** designates that the CombinedFragment represents an assertion. The sequences of the operand of the assertion are the only valid continuations. All other continuations result in an

invalid trace. Assertions are often combined with Ignore or Consider.

Loop

The interaction Operator **loop** designates that the CombinedFragment represents a loop. The **loop** operand will be repeated a number of times.

The Guard may include a lower and an upper number of iterations of the loop as well as a Boolean expression. The semantics is such that a loop will iterate minimum the 'minint' number of times (given by the iteration expression in the guard) and at most the 'maxint' number of times. After the minimum number of iterations have executed, and the boolean expression is false the loop will terminate. The loop construct represent a recursive application of the **seq** operator where the **loop** operand is sequenced after the result of earlier iterations.

The Semantics of Gates

The gates of a CombinedFragment represent the syntactic interface between the CombinedFragment and its surroundings, which means the interface towards other InteractionFragments.

The only purpose of gates is to define the source and the target of messages.

6.4.1.7 Datastore



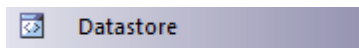
Description

A Datastore is an element used to define permanently stored data. A token of data that enters into a Datastore is stored permanently, updating tokens for data that already exists. A token of data that comes out of a Datastore is a copy of the original data.

Use Object Flow connectors to connect elements (such as Activities) to Datastores, as values and information are being passed between nodes. Selection and transformation behavior, together composing a sort of query, can be specified as to the nature of data access. For instance, selection behavior determines which objects are affected by the connection to the Datastore. Transformation behavior might then further specify the value of an attribute pertaining to a selected object.

To define the behavior of access to a Datastore, attach a note to the Object Flow connector. To do this, right-click on the Object Flow and select the Attach Note or Constraint context menu option. A dialog indicates other flows in the Activity diagram, to which you can attach the note (if the behavior applies to multiple flows). To comply with UML 2, preface behavior with the notation «selection» or «transformation».

Toolbox icon



Learn more

- [Activity Diagram](#) ^[1028]
- [Activity](#) ^[1101]
- [Object Flow](#) ^[1258]

OMG UML Specification:

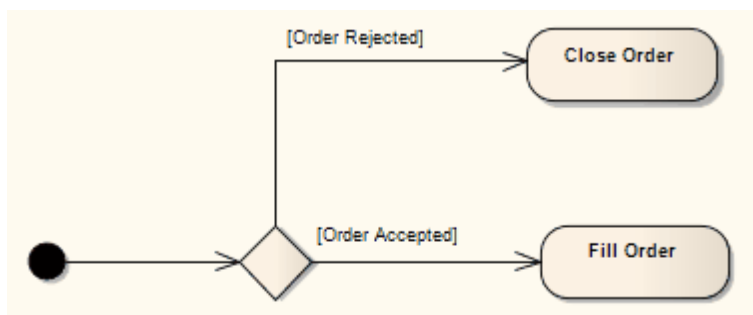
The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 360*) states:

A data store node is a central buffer node for non-transient information... A data store keeps all tokens that enter it, copying them when they are chosen to move downstream. Incoming tokens containing a particular object replace any tokens in the object node containing that object.

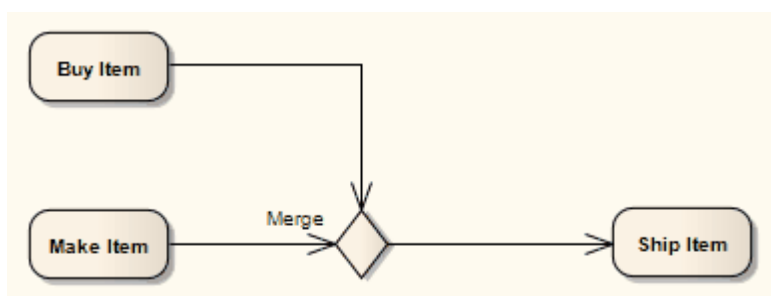
6.4.1.8 Decision**Description**

A Decision is an element of an Activity diagram or Interaction Overview diagram that indicates a point of conditional progression: if a condition is true, then processing continues one way; if not, then another.

This can also be used as a Merge node in that multiple alternative flows can be merged (but not synchronized) to form one flow. The following examples show both of these modes of using the decision element.

Used as a decision:

See *UML Superstructure Specification, v2.1.1, figure 12.77, p. 363*.

Used as a merge:

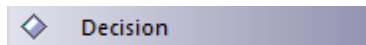
See *UML Superstructure Specification, v2.1.1, figure 12.106, p. 388.*

Notes

- Moving a diagram generally does not affect the location of elements in packages; if you move a diagram out of one package into another, all the elements in the diagram remain in the original package

However, Decision elements are used only within one diagram, have no meaning outside that diagram, and are never re-used in any other diagram; therefore, if you move a diagram containing these elements, they are moved to the new parent package with the diagram

Toolbox icon



Learn more

- [Activity Diagram](#) ^[1028]
- [Interaction Overview Diagram](#) ^[1087]
- [Merge Node](#) ^[1145]

OMG UML Specification:

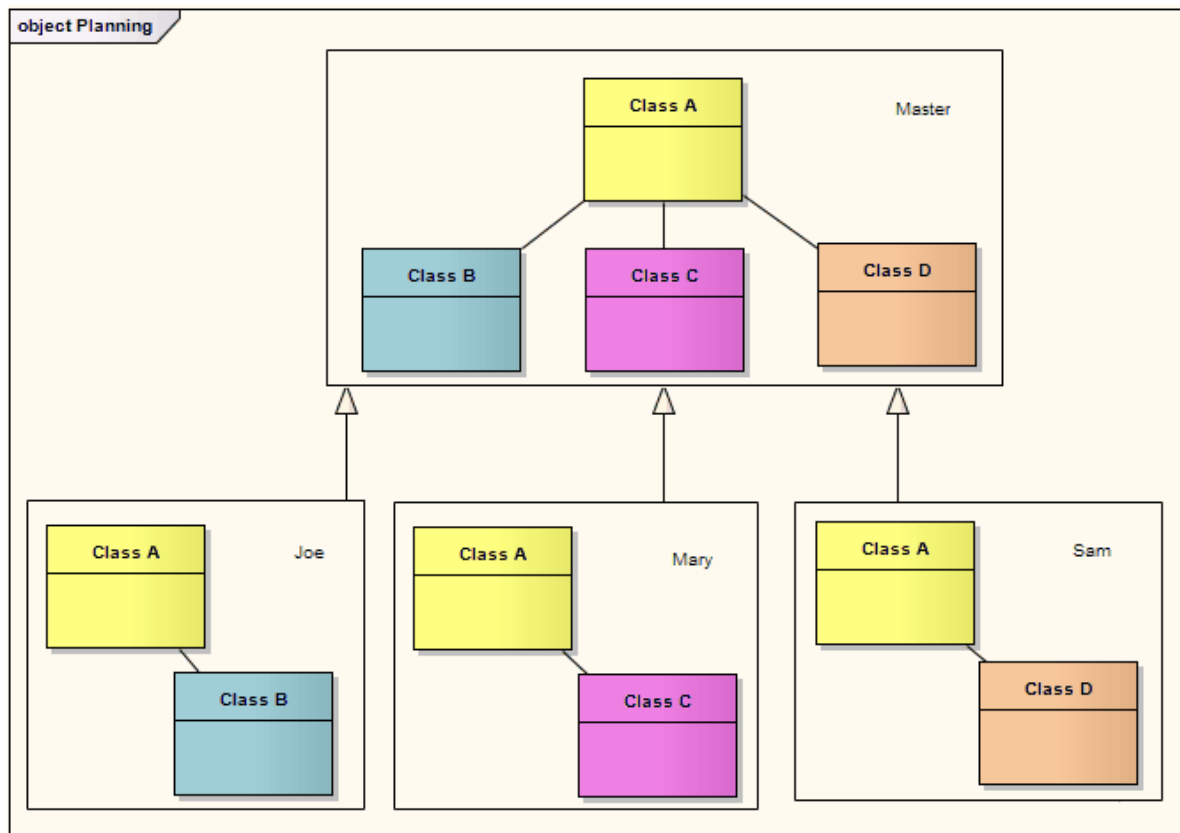
The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 361 (Decision symbol)*) states:

A decision node is a control node that chooses between outgoing flows. A decision node has one incoming edge and multiple outgoing activity edges.

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 387 (Merge symbol)*) also states:

A merge node is a control node that brings together multiple alternate flows. It is not used to synchronize concurrent flows but to accept one among several alternate flows... A merge node has multiple incoming edges and a single outgoing edge.

6.4.1.9 Diagram Frame



Description

A Diagram Frame element is a rendition of a diagram dropped from the Project Browser into another diagram. It is a type of Combined Fragment with the Interaction Operator **ref**. However, it can be created on any type of diagram, and is not created in the same way as other Combined Fragments.

When you drop the diagram from the Project Browser onto the open diagram, a dialog shows providing the following options:

- **Diagram Frame** - a Diagram Frame is inserted into the diagram, containing an image of the dropped diagram
- **Diagram Reference** - an empty frame is inserted with the name of the dropped diagram in the frame label
- **Hyperlink** - a diagram icon is inserted with no frame, and with the parent package and diagram name next to it

In all three cases, the object acts as a hyperlink to the real referenced diagram. You can also define properties for the objects, as for other elements, by right-clicking on the object and selecting the element **Properties** context menu option.

Notes

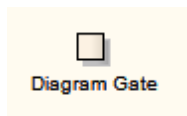
- You can change the size of all three objects, but you cannot reduce a Diagram Frame to less than the size of the enclosed diagram
- You cannot change the diagram within a Diagram Frame; to edit the diagram, double-click within the frame and edit the original diagram
- The Diagram Frame element looks identical to (but is not the same as) a diagram frame border, which you can set automatically on new images of diagrams using the **Tools | Options | Diagram** option, and selecting the appropriate checkboxes in the Diagram Frames panel

These options set frames on print-outs of diagrams, images of diagrams copied to file, and images of diagrams copied to the clipboard; if you paste the image from the clipboard into another diagram, the image initially looks the same as the Diagram Frame element but it is actually a discreet unit that you manipulate using the Image Manager

Learn more

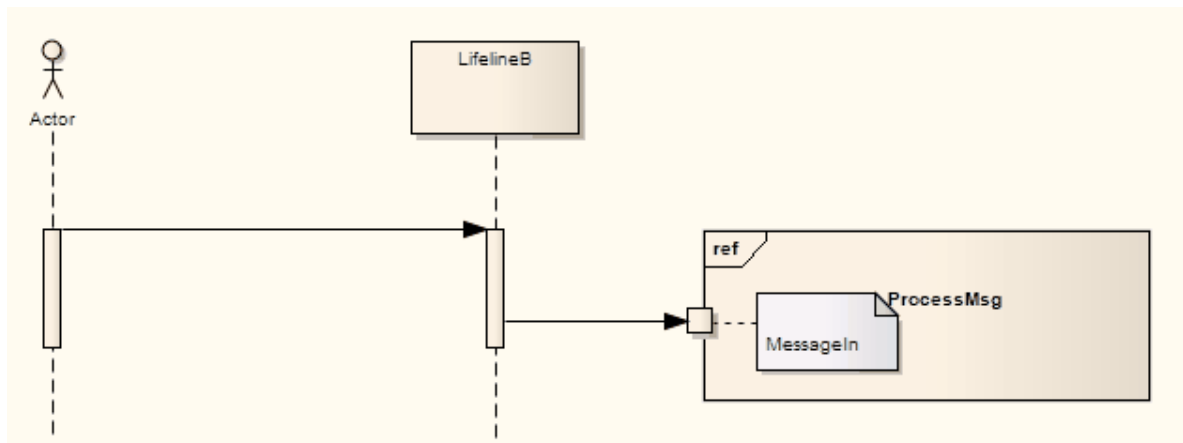
- [Combined Fragment](#)^[1110]
- [Interaction Operator: ref](#)^[1114]

6.4.1.10 Diagram Gate

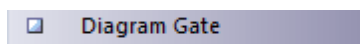


Description

A Diagram Gate is a simple graphical way to indicate the point at which messages can be transmitted into and out of interaction fragments. A fragment might be required to receive or deliver a message; internally, an ordered message reflects this requirement, with a gate indicated on the boundary of the fragment's frame. Any external messages 'synching' with this internal message must correspond appropriately. Gates can appear on Interaction diagrams (Sequence, Timing, Communication or Interaction Overview), interaction occurrences and combined fragments (to specify the expression).



Toolbox icon



Learn more

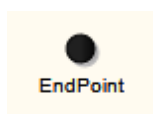
- [Sequence Diagram](#) ^[1074]
- [Timing Diagram](#) ^[1051]
- [Communication Diagram](#) ^[1084]
- [Interaction Overview Diagram](#) ^[1087]
- [Interaction Occurrence](#) ^[1139]
- [Combined Fragment](#) ^[1110]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 480*) states:

A Gate is a connection point for relating a Message outside an InteractionFragment with a Message inside the InteractionFragment ... Gates are connected through Messages. A Gate is actually a representative of an OccurrenceSpecification that is not in the same scope as the Gate. Gates play different roles: we have formal gates on Interactions, actual gates on InteractionUses, expression gates on CombinedFragments.

6.4.1.11 Endpoint

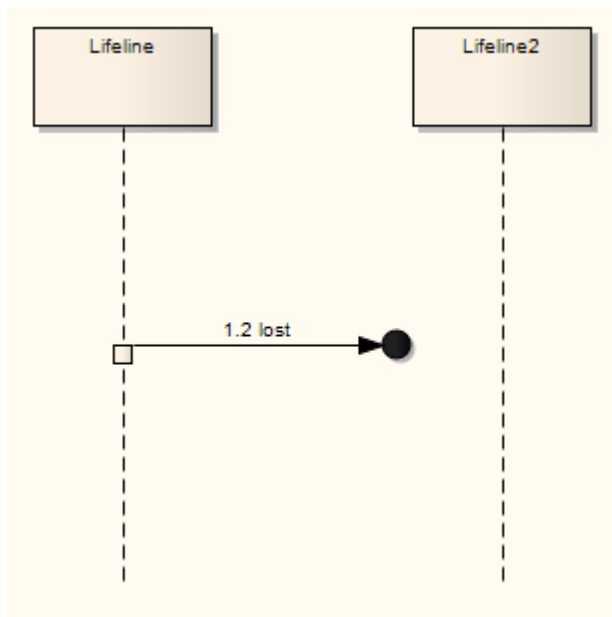


Description

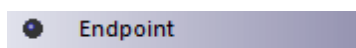
An Endpoint is used in Interaction diagrams (Sequence, Timing, Communication or Interaction Overview) to reflect a lost or found message in sequence. To model this, drag an Endpoint element onto the workspace.

With Sequence diagrams, drag a message from the appropriate lifeline to the Endpoint. With Timing diagrams, the message connecting the lifeline to the Endpoint requires some timing specifications to draw the connection.

The following example depicts a lost message in a Sequence diagram.



Toolbox icon



Learn more

- [Sequence Diagram](#) ^[1074]
- [Timing Diagram](#) ^[1051]
- [Communication Diagram](#) ^[1084]
- [Interaction Overview Diagram](#) ^[1087]

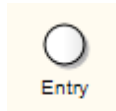
OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 492*) states:

A lost message is a message where the sending event occurrence is known, but there is no receiving event occurrence. We interpret this to be because the message never reached its destination.

A found message is a message where the receiving event occurrence is known, but there is no (known) sending event occurrence. We interpret this to be because the origin of the message is outside the scope of the description. This may, for example, be noise or other activity that we do not want to describe in detail.

6.4.1.12 Entry Point



Description

Entry Point pseudo-states are used to define the beginning of a State Machine. An Entry Point exists for each region, directing the initial concurrent state configuration.

Toolbox icon



Learn more

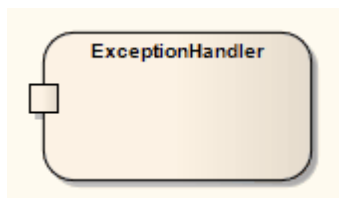
- [State Machine Diagram](#) ^[1032]
- [Pseudo-states](#) ^[1037]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 471) states:

An entry point pseudostate is an entry point of a state machine or composite state. In each region of the state machine or composite state it has a single transition to a vertex within the same region.

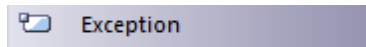
6.4.1.13 Exception



Description

The Exception Handler element defines the group of operations to carry out when an exception occurs. In an Activity diagram, the protected element can contain a set of operations and is connected to the exception handler via an Interrupt Flow connector. Any defined error contained within an element's parts can trigger the flow to move to an exception.

Toolbox icon



Learn more

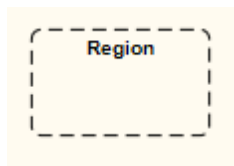
- [Activity Diagram](#) ^[1028]
- [Interrupt Flow](#) ^[1230]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 364*) states:

An exception handler is an element that specifies a body to execute in case the specified exception occurs during the execution of the protected node.

6.4.1.14 Expansion Region



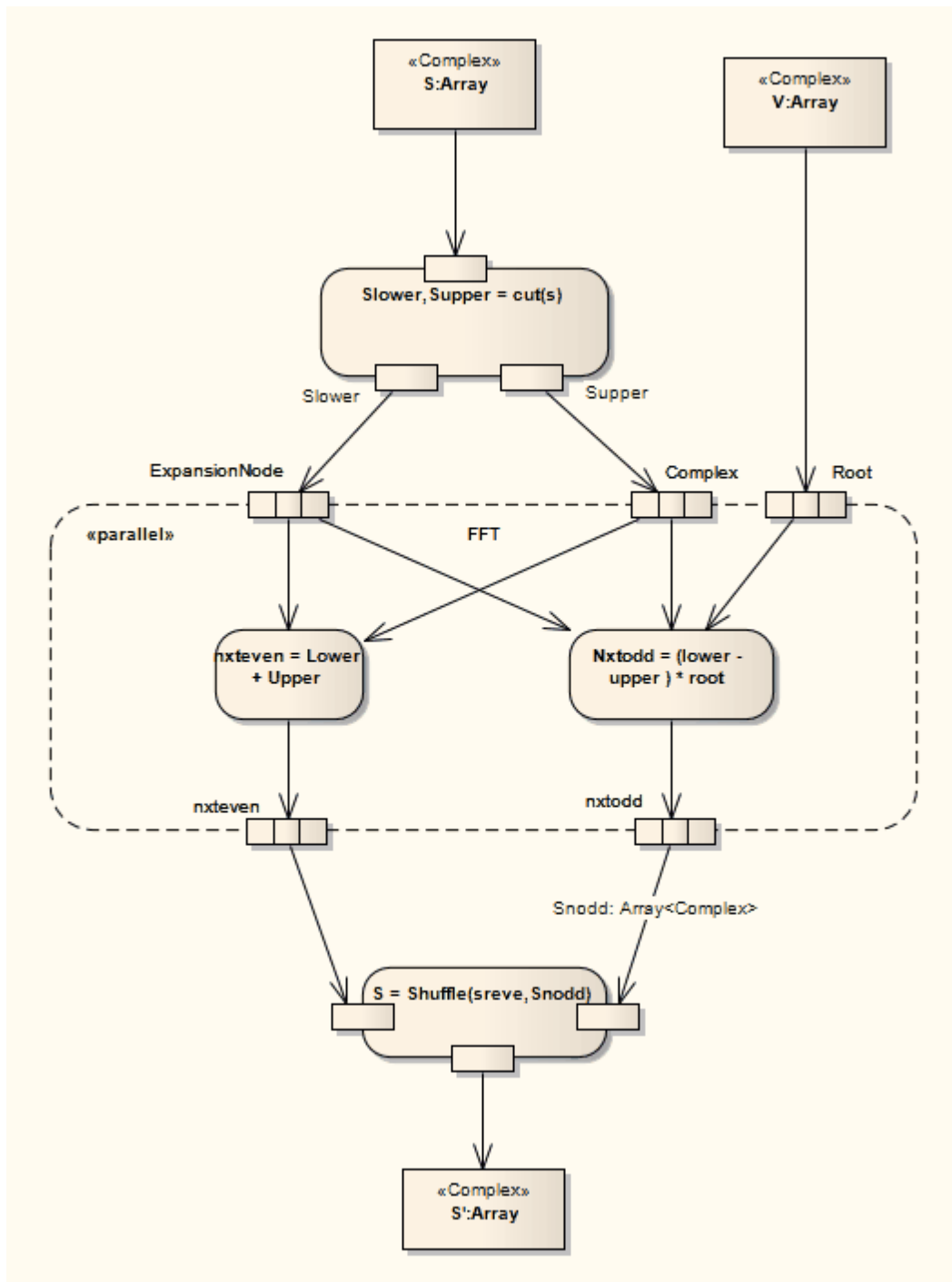
Description

You create an Expansion Region as one variant of a Region (the other is an Interruptible Activity Region).

On an Activity diagram, an Expansion Region surrounds a process to be imposed multiple times on the incoming data, once for every element in the input collection. If there are multiple inputs, the collection sizes must match, and the elements within each collection must be of the same type. Similarly, any outputs must be in the form of a collection matching the size of the inputs.

The concurrency of the Expansion Region's multiple executions can be specified as type parallel, iterative, or stream. Parallel reflects that the elements in the incoming collections can be processed at the same time or overlapping, whereas an iterative concurrency type specifies that execution must occur sequentially. A stream-type Expansion Region indicates that the input and output come in and exit as streams, and that the Expansion Region's process must have some method to support streams.

To modify the mode of an Expansion Region, right-click on it and select the **Properties** context menu option, then select the Advanced page.



See UML Superstructure Specification, v2.1.1, figure 12.87, p. 372.

Toolbox icon



Learn more

- [Activity Diagram](#) ^[1028]
- [Region](#) ^[1150]
- [Interruptible Activity Region](#) ^[1147]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 367*) states:

An expansion region is a structured activity region that executes multiple times corresponding to elements of an input collection.

6.4.1.15 Exit Point



Description

Exit Points are used in Submachine states and State Machines to denote the point where the machine is exited and the transition sourcing this exit point, for Submachines, is triggered. Exit points are a type of pseudo-state used in the State Machine diagram.

Toolbox icon



Learn more

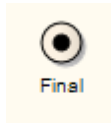
- [State Machine Diagram](#) ^[1032]
- [State Element](#) ^[1152]
- [Submachine State](#) ^[1167]
- [Pseudo-state](#) ^[1037]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1 p. 538*) states:

An exit point pseudostate is an exit point of a state machine or composite state. Entering an exit point within any region of the composite state or state machine referenced by a submachine state implies the exit of this composite state or submachine state and the triggering of the transition that has this exit point as source in the state machine enclosing the submachine or composite state.

6.4.1.16 Final

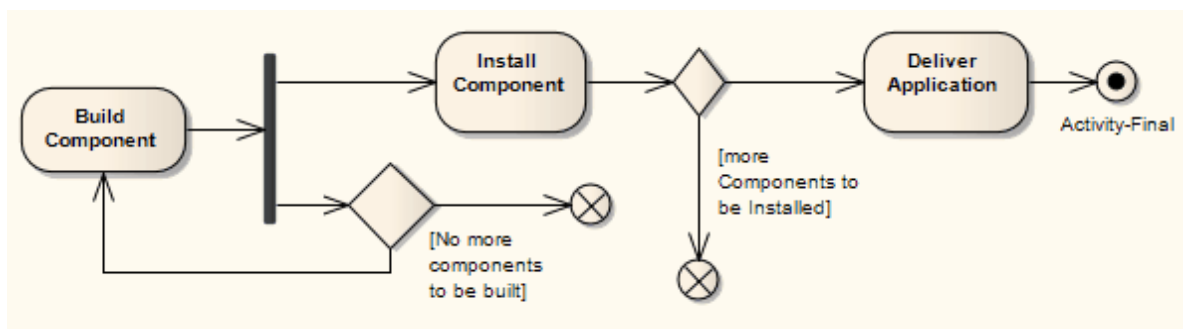


Description

There are two nodes used to define a Final state in an Activity, both defined in UML 2.3 as of type Final Node. The Activity Final element, shown above, indicates the completion of an Activity; upon reaching the Final, all execution in the Activity diagram is aborted. The other type of final node, Flow Final, depicts an exit from the system that has no effect on other executing flows in the Activity.

The following example illustrates the development of an application. The process comes to a Flow Final node when there are no more components to be built; note that the Fork element indicates a concurrent process with the building of new components and installation of completed components. The Flow Final terminates only the sub-process building components. Similarly, only those tokens entering the decision branch for the installation of further components terminate with the connecting Flow Final (that is, stop installing this component, but keep on installing other components). It is only after the Deliver Application activity is completed, after the control flow reaches the Final node, that all flows stop.

The node that initiates a flow is the Initial node.



See *UML Superstructure Specification, v2.1.1, figure 12.91, p. 374.*

Notes

- Moving a diagram generally does not affect the location of elements in packages; if you move a diagram out of one package into another, all the elements in the diagram remain in the original package

However, Final elements are used only within one diagram, have no meaning outside that diagram, and are never re-used in any other diagram; therefore, if you move a diagram containing these elements, they are moved to the new parent package with the diagram

Toolbox icon



Learn more

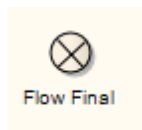
- [Activity Diagram](#) ^[1028]
- [Activity](#) ^[1101]
- [Flow Final](#) ^[1128]
- [Fork](#) ^[1131]
- [Initial](#) ^[1136]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 332*) states:

An activity may have more than one activity final node. The first one reached stops all flows in the activity.

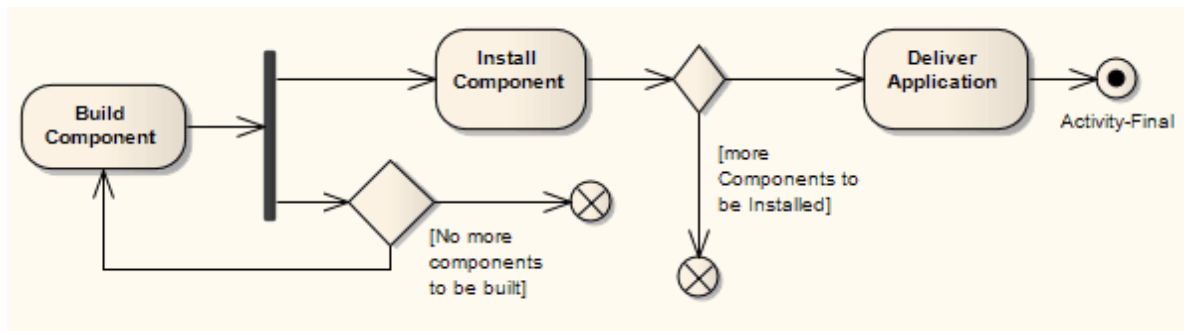
6.4.1.17 Flow Final



Description

There are two nodes used to define a final state in an Activity, both defined in UML 2.3 as of type *Final Node*. The Flow Final element depicts an exit from the system, as opposed to the Activity Final, which represents the completion of the Activity. Only the flow entering the Flow Final node exits the Activity; other flows continue undisturbed.

The following example Activity Diagram illustrates the development of an application. The process comes to a Flow Final node when there are no more components to be built; note that the Fork element indicates a concurrent process with the building of new components and installation of completed components. The Flow Final terminates only the sub-process building components. Similarly, only those tokens entering the decision branch for the installation of further components terminate with the connecting Flow Final (that is, stop installing this component, but keep on installing other components). It is only after the Deliver Application activity is completed, after the control flow reaches the Final node, that all flows stop.



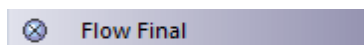
See *UML Superstructure Specification, v2.1.1, figure 12.91, p. 374.*

Notes

- Moving a diagram generally does not affect the location of elements in packages: if you move a diagram out of one package into another, all the elements in the diagram remain in the original package

However, Flow Final elements are used only within one diagram, have no meaning outside that diagram, and are never re-used in any other diagram; therefore, if you move a diagram containing these elements, they are moved to the new parent package with the diagram

Toolbox icon



Learn more

- [Activity Diagram](#) ^[1028]
- [Activity Final](#) ^[1127]
- [Fork](#) ^[1131]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 375*) states:

A flow final destroys all tokens that arrive at it. It has no effect on other flows in the activity.

6.4.1.18 Fork/Join



Description

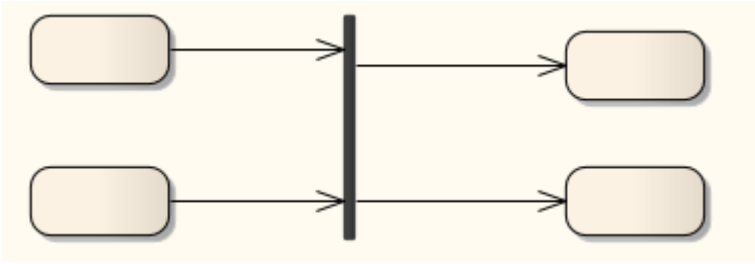
The *Fork/Join* elements have different modes of use, as follows:

- To fork or split the flow into a number of concurrent flows
- To join the flow of a number of concurrent flows
- To both join and fork a number of incoming flows to a number of outgoing flows

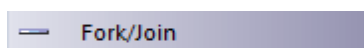
These elements are used in both Activity and State Machine diagrams. With respect to State Machine diagrams, Forks and Joins are used as pseudo-states. Other pseudo-states include history states, entry points and exit points. Forks are used to split an incoming transition into concurrent multiple transitions leading to different target states. Joins are used to merge concurrent multiple transitions into a single transition leading to a single target. They are semantic inverses. To learn more about these individual elements see their specific topics.

Example Diagrams:

Description	Diagram
<p>Fork or split the flow into a number of concurrent flows</p>	
<p>Join the flow of a number of concurrent flows</p>	

Description	Diagram
<p>Join and Fork a number of incoming flows to a number of outgoing flows</p>	

Toolbox icon



Learn more

- [Activity Diagram](#) ^[1028]
- [State Machine Diagram](#) ^[1032]
- [Pseudo-state](#) ^[1037]
- [Fork](#) ^[1131]
- [Join](#) ^[1133]

OMG UML Specification:

Fork

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 376*) states:

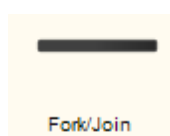
A fork node is a control node that splits a flow into multiple concurrent flows... A fork node has one incoming edge and multiple outgoing edges.

Join

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 381-382*) states:

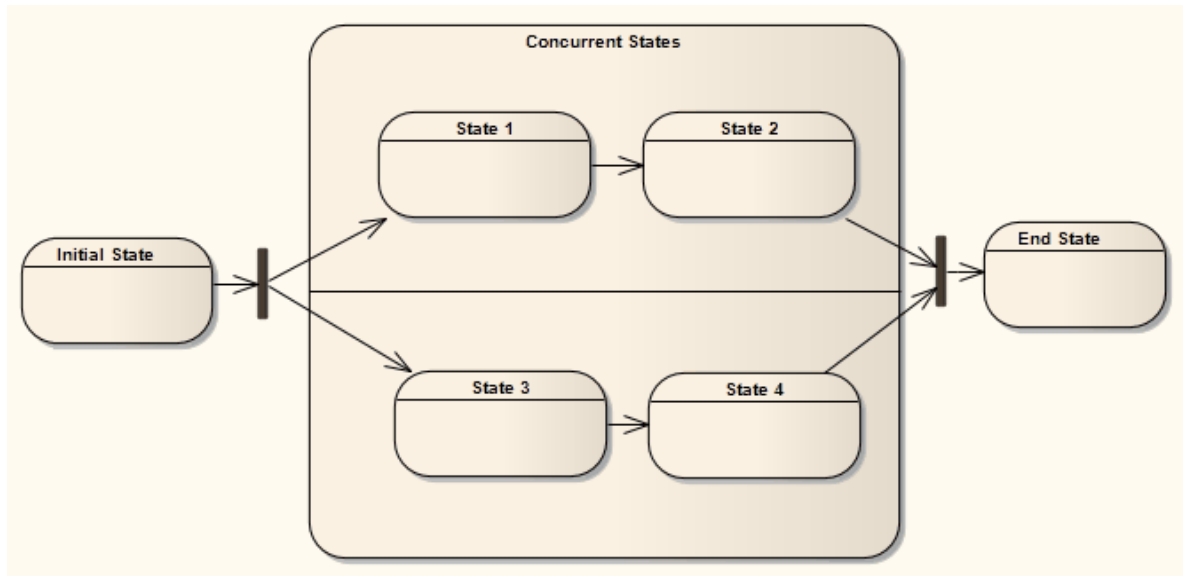
A join node is a control node that synchronizes multiple flows... A join node has multiple incoming edges and one outgoing edge.

6.4.1.18.1 Fork



Description

The Fork element is used in both Activity and State Machine diagrams. With respect to State Machine diagrams, a Fork pseudo-state signifies that its incoming transition comes from a single state, and it has multiple outgoing transitions. These transitions must occur concurrently, requiring the use of concurrent regions, as depicted below in the Composite State. Unlike Choice or Junction pseudo-states, Forks must not have triggers or guards. The following diagram demonstrates a Fork pseudo-state dividing into two concurrent regions, which then return to the End State via the Join pseudo-state.



Learn more

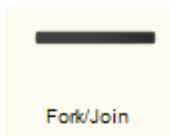
- [Activity Diagram](#) ^[1028]
- [State Machine Diagram](#) ^[1032]
- [Pseudo-state](#) ^[1037]
- [States](#) ^[1152]
- [Choice](#) ^[1108]
- [Junction](#) ^[1142]
- [Join](#) ^[1135]
- [Region](#) ^[1150]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 538*) states:

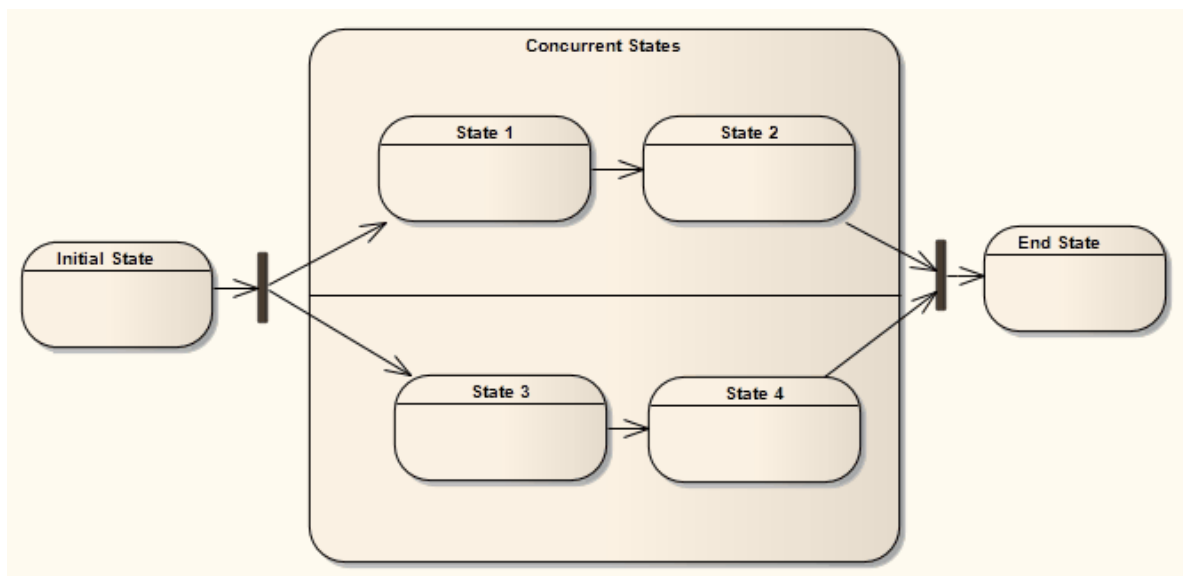
Fork vertices serve to split an incoming transition into two or more transitions terminating on orthogonal target vertices (i.e. vertices in different regions of a composite state). The segments outgoing from a fork vertex must not have guards or triggers.

6.4.1.18.2 Join



Description

The Join element is used by Activity and State Machine diagrams. The above example illustrates a Join transition between Activities. With respect to State Machine diagrams, a Join pseudo-state indicates multiple States concurrently transitioning into the Join and onto a single State. Unlike Choice or Junction pseudo-states, Joins must not have triggers or guards. The following diagram demonstrates a Fork pseudo-state dividing into two concurrent Regions, which then return to the End State via the Join.



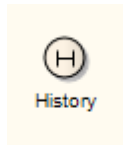
Learn more

- [Activity Diagram](#) ^[1028]
- [State Machine Diagram](#) ^[1032]
- [Pseudo-state](#) ^[1037]
- [States](#) ^[1152]
- [Choice](#) ^[1108]
- [Junction](#) ^[1142]
- [Fork](#) ^[1131]
- [Region](#) ^[1150]

OMG UML Specification:

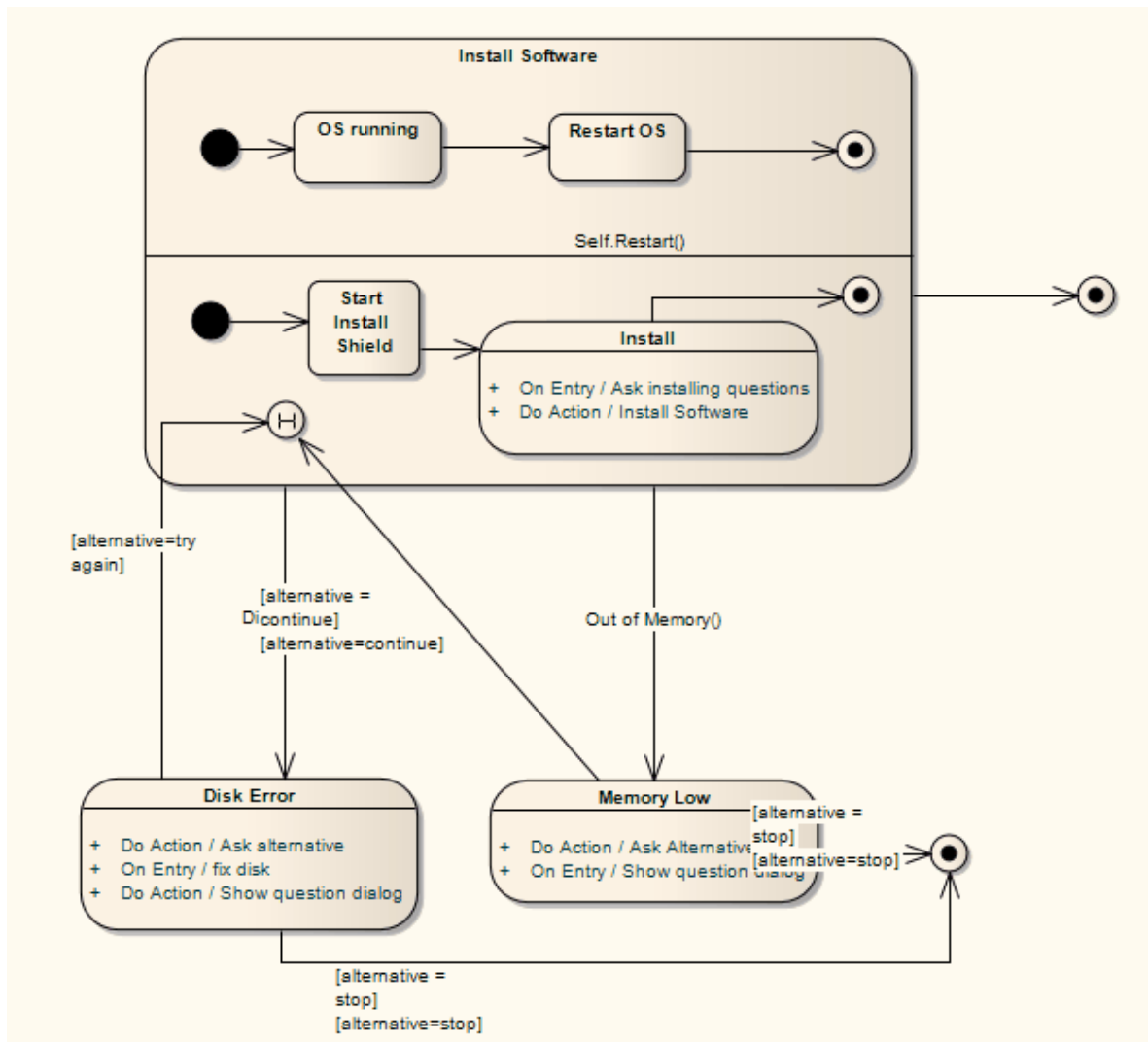
The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 538*) states:

Join vertices serve to merge several transitions emanating from source vertices in different orthogonal regions. The transitions entering a join vertex cannot have guards or triggers.

6.4.1.19 History**Description**

There are two types of History pseudo-state defined in UML: shallow and deep history. A shallow History sub-state is used to represent the most recently active sub-state of a Composite State; this pseudo-state does not recurse into this sub-state's active configuration, should one exist. A single connector can be used to depict the default shallow History state, in case the Composite State has never been entered.

A deep History sub-state, in contrast, reflects the most recent active configuration of the Composite State. This includes active sub-states of all regions, and recurses into those sub-states' active sub-states, should they exist. At most one deep history and one shallow history can dwell within a composite state. You can reassign a shallow History substate as a deep History substate using the Advanced element context menu.



Toolbox icon



Learn more

- [Pseudo-state](#) ^[1037]
- [Composite State](#) ^[1153]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 537*) states:

... *deepHistory* represents the most recent active configuration of the composite state that directly contains this pseudostate (e.g., the state configuration that was active when the composite state was last exited). A

composite state can have at most one deep history vertex. At most one transition may originate from the history connector to the default deep history state. This transition is taken in case the composite state had never been active before. Entry actions of states entered on the path to the state represented by a deep history are performed.

... shallowHistory represents the most recent active substate of its containing state (but not the substates of that substate). A composite state can have at most one shallow history vertex. A transition coming into the shallow history vertex is equivalent to a transition coming into the most recent active substate of a state. At most one transition may originate from the history connector to the default shallow history state. This transition is taken in case the composite state had never been active before. Entry actions of states entered on the path to the state represented by a shallow history are performed.

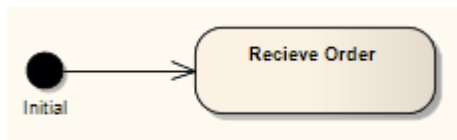
6.4.1.20 Initial



Description

The Initial element is used by Activity and State Machine diagrams. In Activity diagrams, it defines the start of a flow when an Activity is invoked. With State Machines, the Initial element is a pseudo-state used to denote the default state of a Composite State; there can be one Initial vertex in each Region of the Composite State.

This simple example shows the start of a flow to receive an order.



See *UML Superstructure Specification, v2.1.1, Figure 12.97, p. 378.*

The activity flow is completed by a Final or Flow Final node.

Notes

- Moving a diagram generally does not affect the location of elements in packages; if you move a diagram out of one package into another, all the elements in the diagram remain in the original package

However, Initial elements are used only within one diagram, have no meaning outside that diagram, and are never re-used in any other diagram; therefore, if you move a diagram containing these elements, they are moved to the new parent package with the diagram

Toolbox icon



Learn more

- [Activity Diagram](#) ^[1028]
- [State Machine Diagram](#) ^[1032]
- [Activity](#) ^[1101]
- [Pseudo-state](#) ^[1037]
- [Composite State](#) ^[1153]
- [Region](#) ^[1150]
- [Final](#) ^[1127]
- [Flow Final](#) ^[1128]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 537*) states:

An initial pseudostate represents a default vertex that is the source for a single transition to the default state of a composite state. There can be at most one initial vertex in a region. The outgoing transition from the initial vertex may have a behavior, but not a trigger or guard.

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 378*) also states:

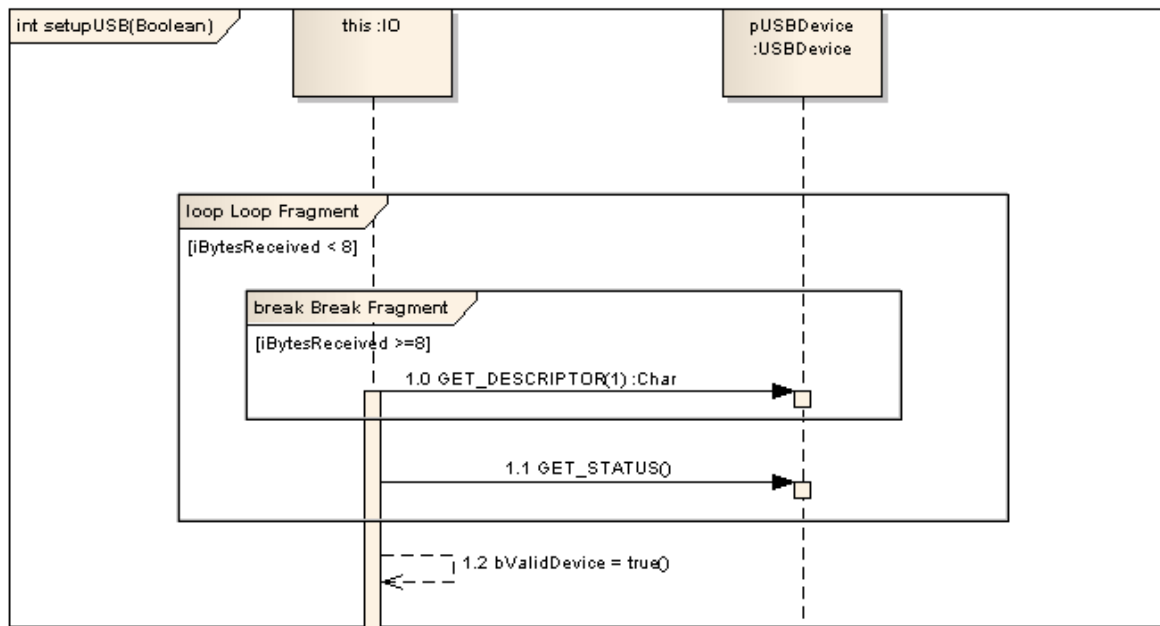
An initial node is a control node at which flow starts when the activity is invoked.

6.4.1.21 Interaction

Description

An *Interaction* element is used to describe a system, representing its interactions at varying levels of detail, for review not only by design professionals but also by end users and stakeholders. An Interaction element can contain the following types of diagram:

- Sequence
- Interaction Overview
- Communication
- Timing

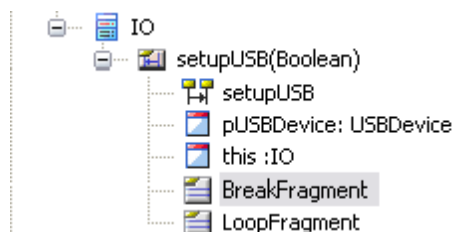


An Interaction element in Enterprise Architect is treated as a behavior of the classifier it is encapsulated within. It can have parameters and return types, which are modeled using the Behavior tab of the Interaction element's Properties dialog. The element is interpreted as a method of the containing Class in the generated code (see the *Generate Code From Behavioral Model* topic).

An Interaction element can also be set as the classifier for an *Interaction Occurrence* in a Sequence diagram, or for a *Call Behavior Action* in an Activity diagram. Establishing such an association (between a behavior and a behavior call) facilitates adding arguments that can be individually mapped to the associated behavior's parameters.

Notes

- The behavioral code generation engine expects the Sequence diagram and all its associated messages and interaction fragments to be encapsulated within an Interaction element (such as `setupUSB` in the example below)



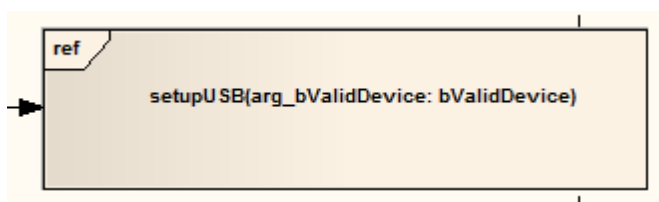
(The IO Class shown above is available in the EAExample model, under **Systems Engineering Model | Implementation Model | Software**)

Learn more

- [Sequence Diagram](#) ^[1074]

- [Interaction Overview Diagram](#) ^[1087]
- [Communication Diagram](#) ^[1084]
- [Timing Diagram](#) ^[1051]
- [Generate Code From Behavioral Model](#) ^[1778]
- [Interaction Occurrence](#) ^[1139]
- [Call Behavior Action](#) ^[1095]
- [Associate with Different Behavior](#) ^[911]
- [Arguments](#) ^[912]

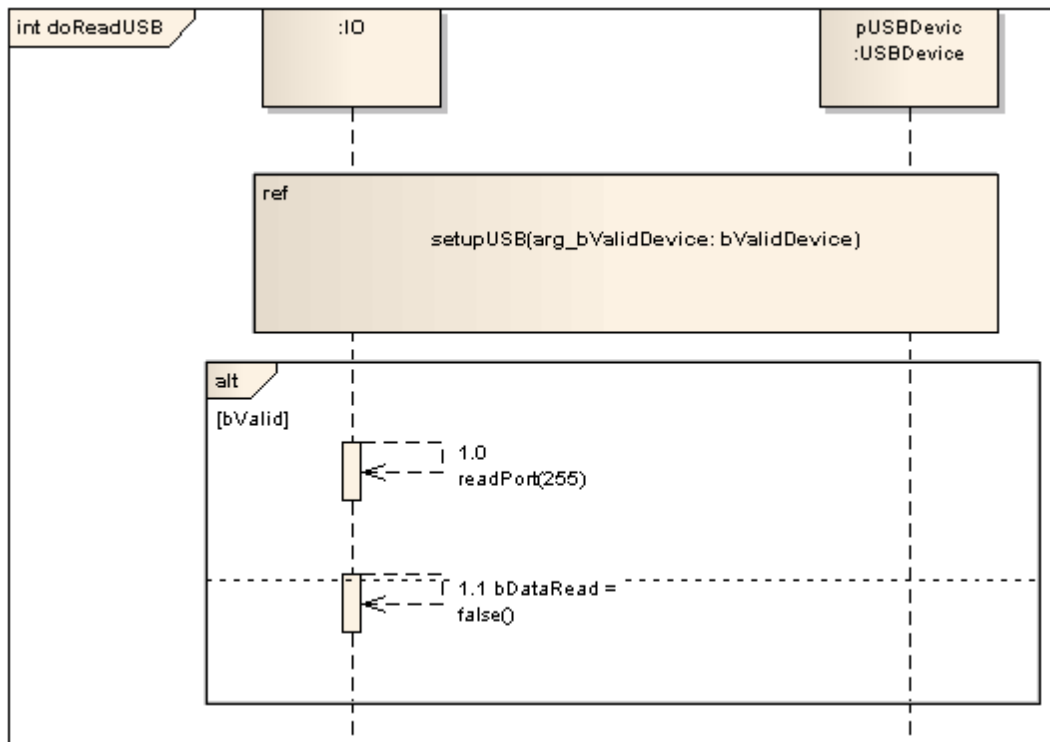
6.4.1.22 Interaction Occurrence



Description

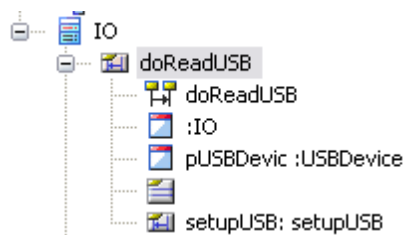
An *Interaction Occurrence* (or *InteractionUse*) is a reference to an existing *Interaction* (Sequence) diagram. Interaction Occurrences are visually represented by a frame, with **ref** in the frame's title space. The diagram name is indicated in the frame contents. To create an Interaction Occurrence, simply open a Sequence diagram (preferably contained within an Interaction element) and drag another Sequence diagram (also preferably contained within an Interaction element) into its workspace. A dialog displays, providing configuration options. The resulting Interaction Occurrence acts as an invocation of the original Interaction. You use the Call tab of the element Properties dialog to set up the actual arguments of the Interaction and also to change to a different associated Interaction element.

The following figure illustrates the use of an Interaction Occurrence in another Interaction (Sequence) diagram. You can display the sequence represented by the Interaction Occurrence by double-clicking on the element.



Notes

- The behavioral code generation engine expects the Sequence diagram and all its associated messages and interaction fragments to be encapsulated within an Interaction element (such as `doReadUSB` in the example below)



Learn more

- [Sequence Diagram](#) ^[1074]
- [Interaction element](#) ^[1137]
- [Behavior Calls Invocations](#) ^[910]
- [Arguments](#) ^[912]

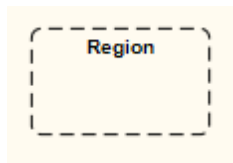
OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 423*) refers to an Interaction Occurrence as an **InteractionUse**, and states:

An InteractionUse refers to an Interaction. The InteractionUse is a shorthand for copying the contents of the referred Interaction where the InteractionUse is. To be accurate the copying must take into account substituting parameters with arguments and connect the formal gates with the actual ones.

It is common to want to share portions of an interaction between several other interactions. An InteractionUse allows multiple interactions to reference an interaction that represents a common portion of their specification.

6.4.1.23 Interruptible Activity Region

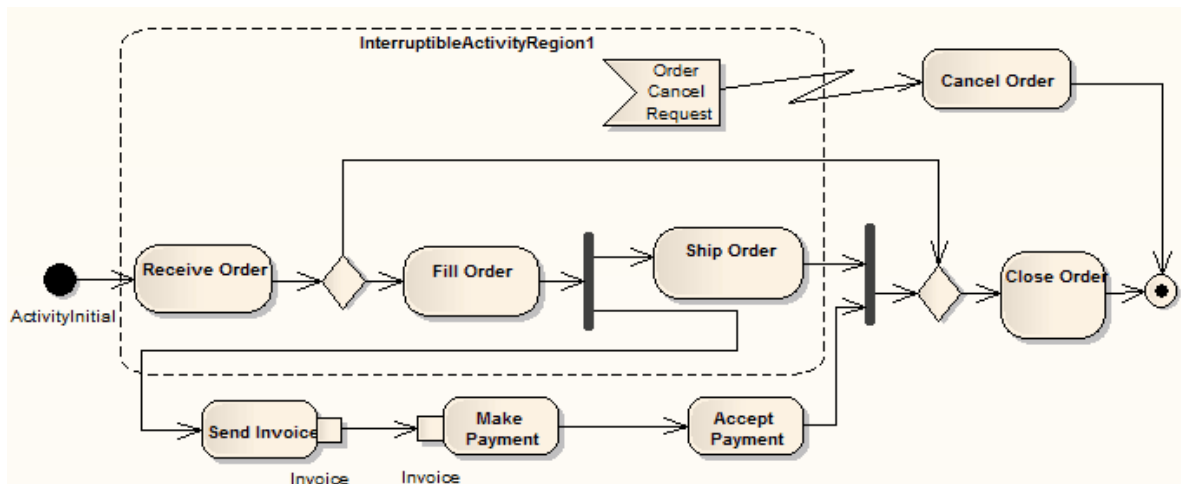


Description

You create an Interruptible Activity Region as one variant of a Region (the other is an Expansion Region).

In an Activity diagram, an Interruptible Activity Region surrounds a group of Activity elements, all affected by certain interrupts in such a way that all tokens passing within the region are terminated should the interruption(s) be raised. Any processing occurring within the bounds of an Interruptible Activity Region is terminated when a flow is instigated across an interrupt flow to an external element.

The example below illustrates that an order cancellation kills any processing of the order at the receipt, filling or shipping stage.



See *UML Superstructure Specification, v2.1.1, figure 12.100, p. 381*.

Toolbox icon



Learn more

- [Activity diagram](#) ^[1028]
- [Activity](#) ^[1101]
- [Region](#) ^[1150]
- [Expansion Region](#) ^[1124]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 380*) states:

An interruptible region contains activity nodes. When a token leaves an interruptible region via edges designated by the region as interrupting edges, all tokens and behaviors in the region are terminated.

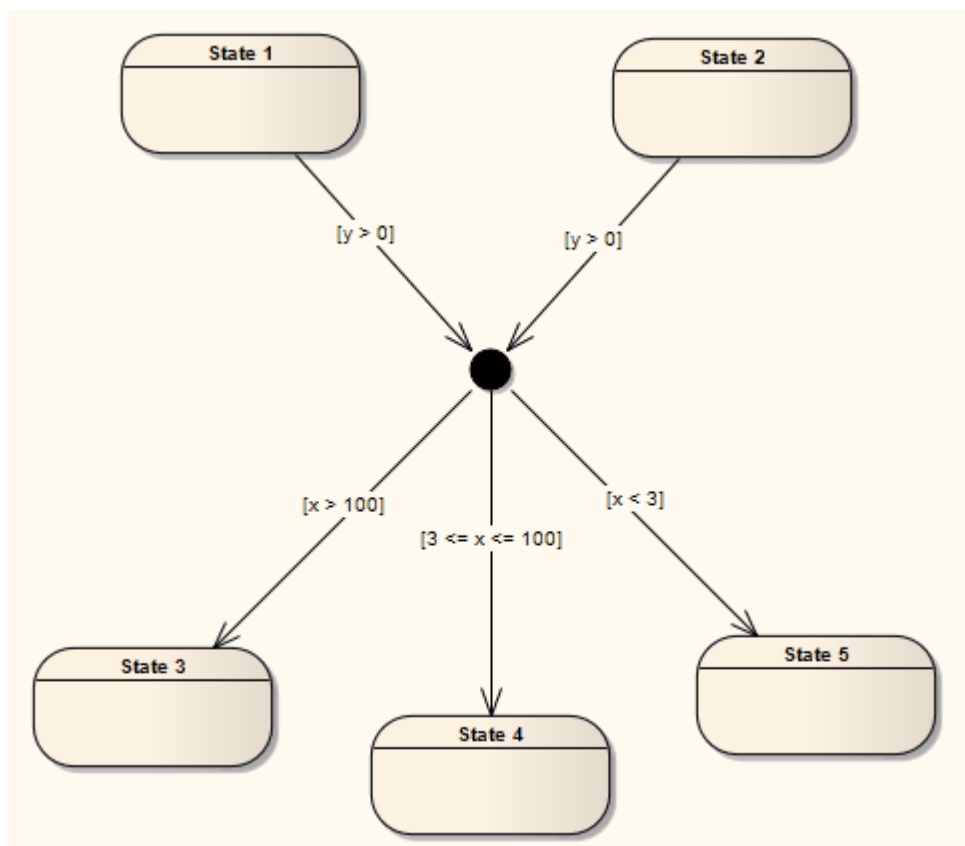
6.4.1.24 Junction



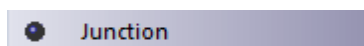
Description

Junction pseudo-states are used to design complex transitional paths in State Machine diagrams. A Junction can be used to combine or merge multiple paths into a shared transition path. Alternatively, a Junction can split an incoming path into multiple paths, similar to a Fork pseudostate. Unlike Forks or Joins, Junctions can apply guards to each incoming or outgoing transition, such that if the guard expression is false, the transition is disabled.

The following example illustrates how guards can be applied to transitions coming into or out of a Junction pseudo-state.



Toolbox icon



Learn more

- [State Machine Diagram](#) ^[1032]
- [Pseudo-states](#) ^[1037]
- [Fork](#) ^[1137]
- [Join](#) ^[1135]

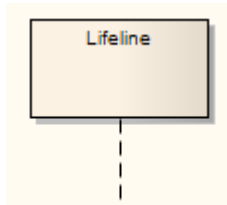
OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 538*) states:

... junction vertices are semantic-free vertices that are used to chain together multiple transitions. They are used to construct compound transition paths between states. For example, a junction can be used to converge multiple incoming transitions into a single outgoing transition representing a shared transition path (this is known as a merge). Conversely, they can be used to split an incoming transition into multiple outgoing transition segments with different guard conditions. This realizes a static conditional branch. (In the latter case, outgoing transitions whose guard conditions evaluate to false are disabled. A predefined guard

denoted "else" may be defined for at most one outgoing transition. This transition is enabled if all the guards labeling the other transitions are false.) Static conditional branches are distinct from dynamic conditional branches that are realized by choice vertices.

6.4.1.25 Lifeline



Description

A Lifeline is an individual participant in an interaction (that is, Lifelines cannot have multiplicity). A Lifeline represents a distinct connectable element. To specify that representation within Enterprise Architect, right-click on the Lifeline and select the **Advanced | Instance Classifier** context menu option. The Select <Item> dialog displays which you use to locate the required project classifiers.

Lifelines are available in Sequence diagrams. There are different Lifeline elements for Timing diagrams (State Lifeline and Value Lifeline); however, although the representation differs between the two diagram types, the meaning of the Lifeline is the same.

Toolbox icon



Learn more

- [Instance Classifier](#) ^[880]
- [Sequence Diagram](#) ^[1074]
- [Timing diagram](#) ^[1051]
- [State Lifeline](#) ^[1158]
- [Value Lifeline](#) ^[1177]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p.489*) states:

A lifeline represents an individual participant in the Interaction. While Parts and StructuralFeatures may have multiplicity greater than 1, Lifelines represent only one interacting entity.

Lifeline is a specialization of NamedElement.

If the referenced ConnectableElement is multivalued (i.e. has a multiplicity > 1), then the Lifeline may have an expression (the 'selector') that specifies which particular part is represented by this Lifeline. If the selector is omitted this means that an arbitrary representative of the multivalued ConnectableElement is chosen.

6.4.1.26 Merge



Description

A Merge Node brings together a number of alternative flow paths in Activity, Analysis and Interaction Overview diagrams. For example, if a Decision is used after a Fork, the two flows coming out of the Decision must be merged into one before going to a Join; otherwise, the Join waits for both flows, only one of which arrives.

A Merge Node has multiple incoming edges and a single outgoing edge. The edges coming into and out of a Merge Node must be either all object flows or all control flows.

Toolbox icon



Learn more

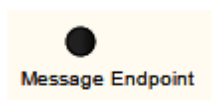
- [Activity Diagram](#) ^[1028]
- [Analysis Diagram](#) ^[1509]
- [Interaction Overview Diagram](#) ^[1087]
- [Decision](#) ^[1117]
- [Fork](#) ^[1131]
- [Join](#) ^[1133]
- [Object flow](#) ^[1250]
- [Control flow](#) ^[1219]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 387*) states:

A merge node is a control node that brings together multiple alternate flows. It is not used to synchronize concurrent flows but to accept one among several alternate flows.

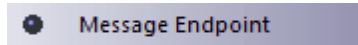
6.4.1.27 Message Endpoint



Description

A Message Endpoint element defines the termination of a State or Value Lifeline in a Timing diagram.

Toolbox icon



Learn more

- [Timing Diagram](#) ^[1051]
- [State Lifeline](#) ^[1156]
- [Value Lifeline](#) ^[1177]

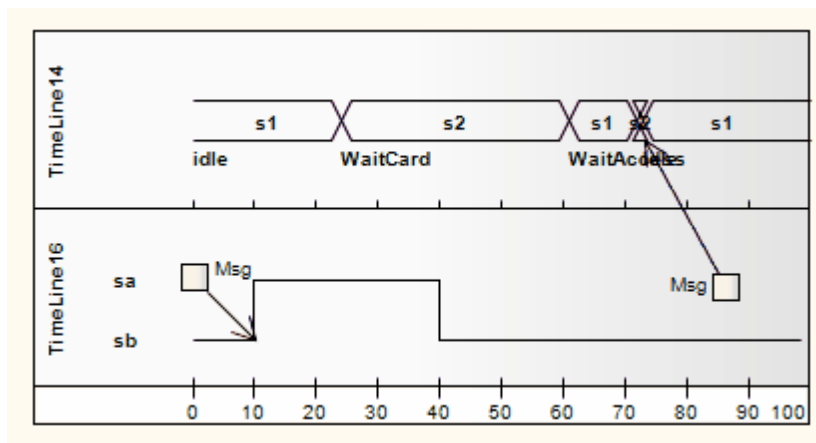
6.4.1.28 Message Label



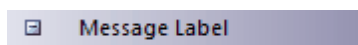
Description

A Message Label is an alternative way of denoting Messages between Lifelines, which is useful for 'uncluttering' Timing diagrams strewn with messages. To indicate a Message between Lifelines, draw a connector from the source Lifeline into a Message Label. Next, draw a connector from another Message Label to the target Lifeline. Note that the label names must match to reflect that the message occurs between the two Message Labels.

The following diagram illustrates how Message Labels are used to construct a message between Lifelines.



Toolbox icon



Learn more

- [Timing diagram](#)^[1051]
- [Messages](#)^[1245]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 518*) states:

Labels are only notational shorthands used to prevent cluttering of the diagrams with a number of messages crisscrossing the diagram between Lifelines that are far apart. The labels denote that a Message may be disrupted by introducing labels with the same name.

6.4.1.29 Note



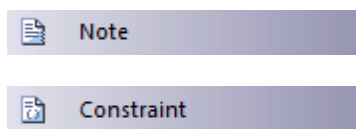
Description

A Note element is a textual annotation that can be attached to a set of elements of any other type. The attachment is created separately, using a Notelink connector. Both Note and Notelink are available in any Enterprise Architect diagram, through the Common pages of the Toolbox.

A Note is also called a Comment.

A Constraint is a form of Note, identifying a constraint on other elements. As for a Note, you can connect the Constraint element to other elements using a Notelink connector. This element is just a means of documenting the fact that there are constraints; it has no impact on the other elements. You define the types of constraint in the project reference data, apply them to the element in the element Properties dialog, and manage them through the Scenarios & Requirements window.

Toolbox icon



Learn more

- [Notelink Connector](#)^[1245]
- [Constraint Types](#)^[997]
- [Scenarios & Requirements Window](#)^[879]

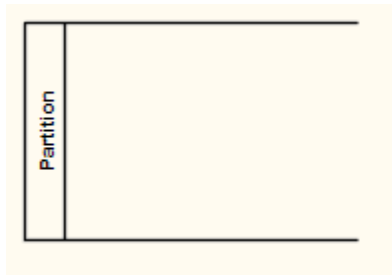
OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 59*) states:

A comment gives the ability to attach various remarks to elements. A comment carries no semantic force, but may contain information that is useful to a modeler.

A comment can be owned by any element.

6.4.1.30 Partition



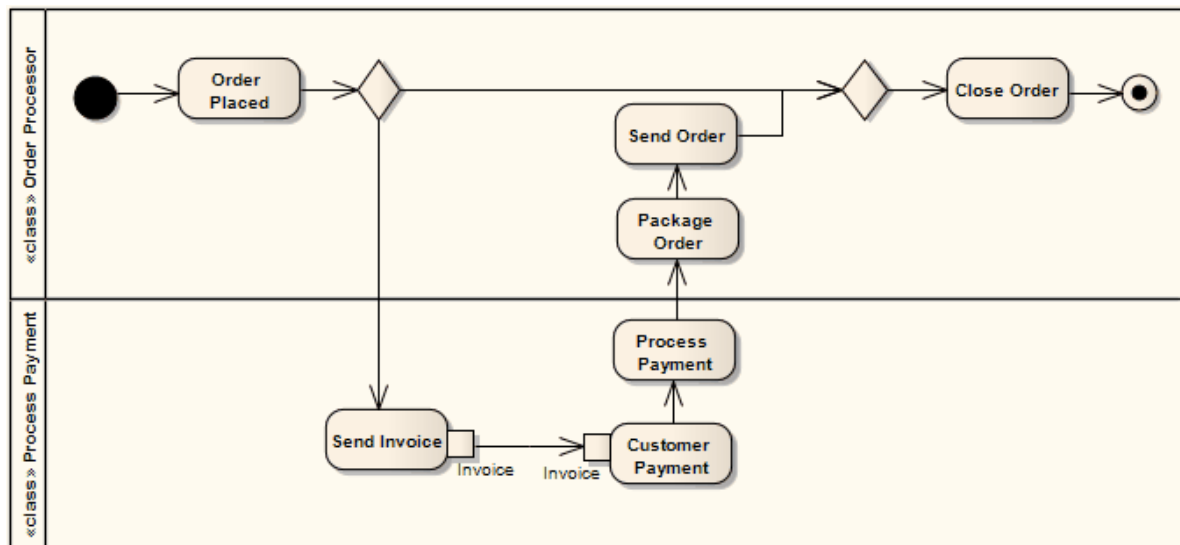
Description

Enterprise Architect supports two types of Activity Partition:

- The Activity Partition feature, which is used to logically organize an Activity element
- The Activity Partition element, described in this topic, which is used to logically organize an Activity diagram

In effect, these are the same. They partition the Actions of the Activity without affecting the token flow, helping to structure the view or parts of the Activity.

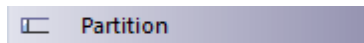
The following example depicts the partitioning between the Classes *Process Payment* and *Order Processor*.



The Partition orientation defaults to horizontal. To turn it into a vertical Partition, right-click on it and select the **Advanced | Vertical Partition** context menu option.

You can neatly align and join the Activity Partitions on a diagram using the element context menu **Dockable** option. For Partitions, the option defaults to selected.

Toolbox icon



Learn more

- [Activity Diagram](#)^[1028]
- [Activity Element](#)^[1101]
- [Activity Partition Feature](#)^[1105]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 341*) states:

Partitions divide the nodes and edges to constrain and show a view of the contained nodes. Partitions can share contents. They often correspond to organizational units in a business model. They may be used to allocate characteristics or resources among the nodes of an activity.

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 341*) also states:

An activity partition is a kind of activity group for identifying actions that have some characteristic in common.

6.4.1.31 Receive



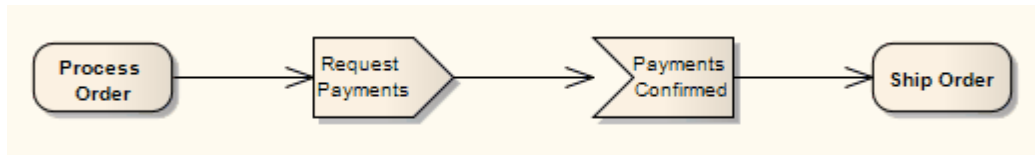
Description

A Receive element is used to define the acceptance or receipt of a request, in an Activity diagram. Movement from a Receive element occurs only once receipt is fulfilled according to its specification. The Receive element comes in two forms:

- *Accept Event Action* element (pennant shape)
- *Accept Time Event Action* element (hourglass shape)

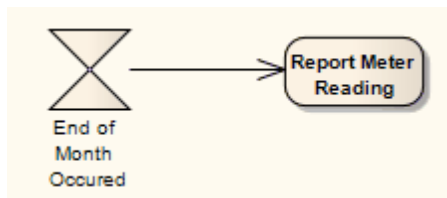
The following example reflects a payment process on an order. Upon receiving the payment (from *Request*

Payments, a Send element), the payment is confirmed and the flow continues to ship the order.



See *UML Superstructure Specification, v2.1.1, figure 12.26, p. 312.*

To depict an Accept Time Event, use the standard Receive element from the Toolbox. Right-click on this element, and select the **Advanced | Accept Time Event** context menu option. The following example shows the hourglass-shaped *Accept Time Event Action*:



See *UML Superstructure Specification, v2.1.1, figure 12.27, p. 312.*

Toolbox icon



Learn more

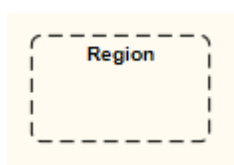
- [Activity Diagram](#) ^[1028]
- [Send Element](#) ^[1151]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 239*) states:

AcceptEventAction is an action that waits for the occurrence of an event meeting specified conditions.

6.4.1.32 Region



Description

Enterprise Architect supports two types of *Region* element:

- Expansion Region
- Interruptible Activity Region

When you add a Region element to an Activity diagram, a prompt displays to enable you to specify the Region type.

Toolbox icon



Learn more

- [Activity Diagram](#)^[1028]
- [Expansion Region](#)^[1124]
- [Interruptible Activity Region](#)^[1141]

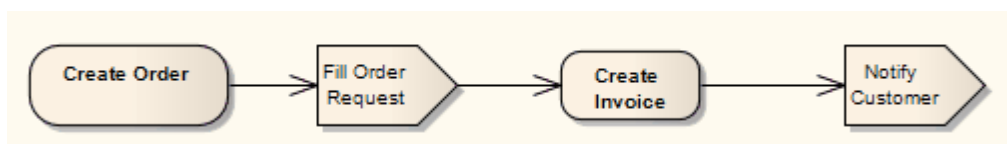
6.4.1.33 Send



Description

The Send element is used to depict the action of sending a signal, in an Activity diagram. It is the opposite of a Receive element.

The following example shows an order being processed, where a signal is sent to fill the processed order and, upon creation of the resulting invoice, a notification is sent to the customer.



See *UML Superstructure Specification, v2.1.1, figure 12.132, p. 408.*

Toolbox icon



Learn more

- [Activity Diagram](#) ^[1028]
- [Receive](#) ^[1149]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 285*) states:

SendObjectAction is an action that transmits an object to the target object, where it may invoke behavior such as the firing of state machine transitions or the execution of an activity. The value of the object is available to the execution of invoked behaviors. The requestor continues execution immediately. Any reply message is ignored and is not transmitted to the requestor.

6.4.1.34 State



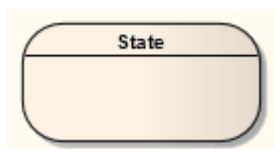
Description

A *State* represents a situation where some invariant condition holds; this condition can be static (waiting for an event) or dynamic (performing a set of activities). State modeling is usually related to Classes, and describes the enableable states a Class or element can be in and the transitions that enable the element to move there. There are two types of State: *Simple States* and *Composite States*, both created from the State element from the Toolbox.


Furthermore, there are pseudo-states, resembling some aspect of a State but with a pre-defined implication. Pseudo-states model complex transitional paths, and classify common State Machine behavior.

You can define entry, internal and exit actions for a State using operations.

If a State element has features such as attributes or operations, the depiction of the element in a diagram has a line under the element name. This line persists if the features are hidden. The line also displays if the **Show State Compartment** checkbox is selected on the Objects page of the Options dialog (**Tools | Options | Objects**).



Toolbox icon

 State**Learn more**

- [State Machine Diagram](#) ^[1032]
- [Class](#) ^[1181]
- [Pseudo-states](#) ^[1037]
- [Operations](#) ^[901]
- [Composite State](#) ^[1153]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 546*) states:

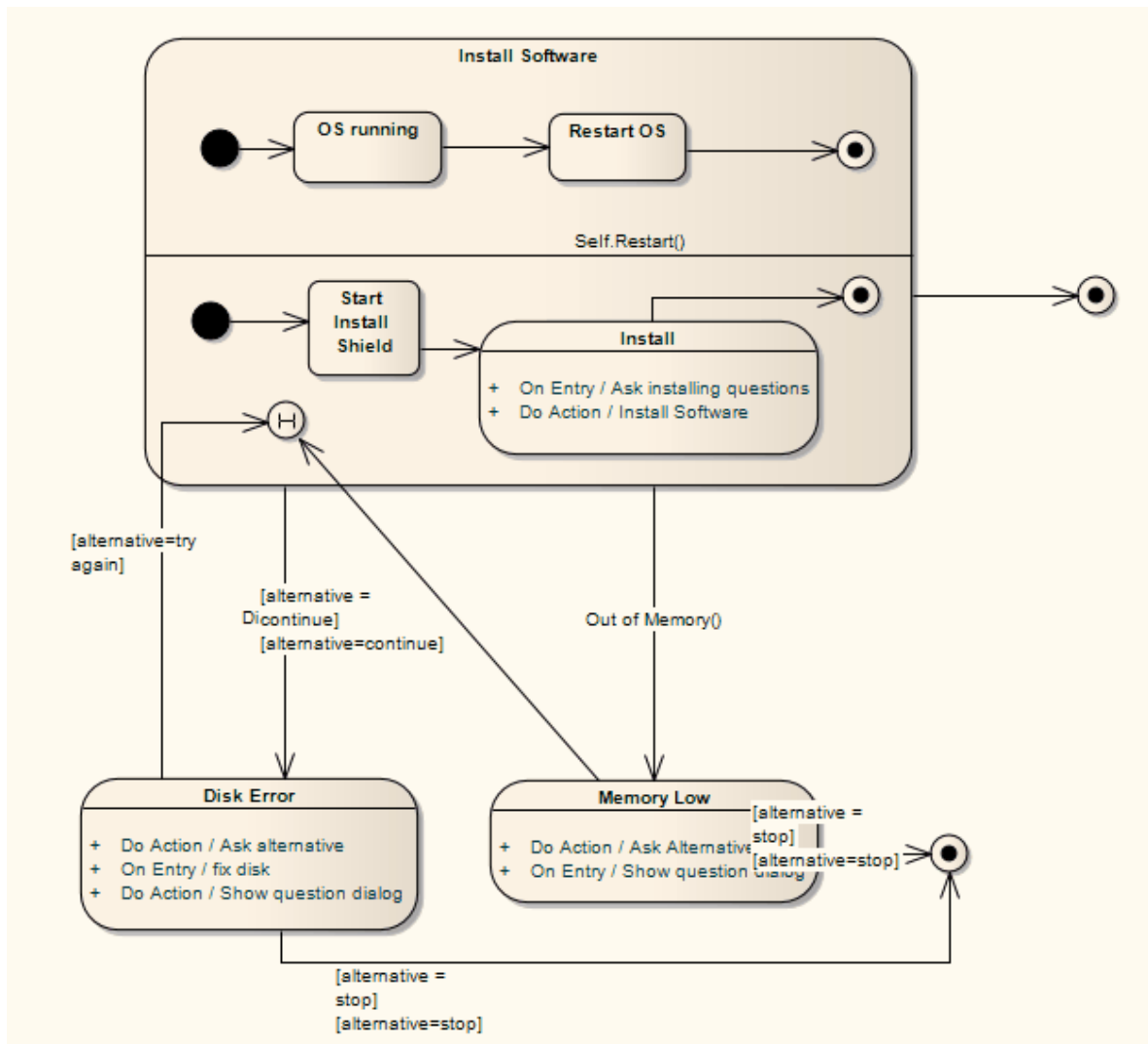
A state models a situation during which some (usually implicit) invariant condition holds. The invariant may represent a static situation such as an object waiting for some external event to occur. However, it can also model dynamic conditions such as the process of performing some activity (i.e., the model element under consideration enters the state when the activity commences and leaves it as soon as the activity is completed).

6.4.1.34.1 Composite State**Description**

Composite States are composed within the State Machine diagram by expanding a State element, adding Regions if applicable, and dragging further State elements, related elements and connectors within its boundaries. The internal State elements are then referred to as Sub-states.

(You can also define a State element, as with many other types of element, as a composite element; this then has a hyperlink to a child diagram that can be another State Machine diagram or other type of diagram elsewhere in the model.)

Composite States can be orthogonal, if Regions are created. If a Composite State is orthogonal, its entry denotes that a single Sub-state is concurrently active in all Regions. The hierarchical nesting of Composite States, coupled with Region use, generates a situation of multiple States concurrently active; this situation is referred to as the active State configuration.



Learn more

- [State Machine Diagram](#) ^[1032]
- [State Element](#) ^[1152]
- [Regions](#) ^[1037]
- [Composite Element](#) ^[825]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 478) states:

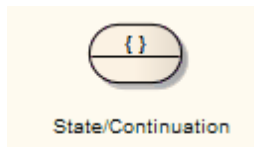
A composite state either contains one region or is decomposed into two or more orthogonal regions. Each region has a set of mutually exclusive disjoint subvertices and a set of transitions. A given state may only be decomposed in one of these two ways.

Any state enclosed within a region of a composite state is called a substate of that composite state. It is called a direct substate when it is not contained by any other state; otherwise it is referred to as an indirect substate.

Each region of a composite state may have an initial pseudostate and a final state. A transition to the enclosing state represents a transition to the initial pseudostate in each region. A newly-created object takes its topmost default transitions, originating from the topmost initial pseudostates of each region.

A transition to a final state represents the completion of activity in the enclosing region. Completion of activity in all orthogonal regions represents completion of activity by the enclosing state and triggers a completion event on the enclosing state. Completion of the topmost regions of an object corresponds to its termination.

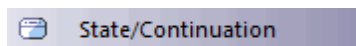
6.4.1.35 State/Continuation



Description

The State/Continuation element serves two different purposes for Interaction (Sequence) diagrams, as State Invariants and Continuations. Enterprise Architect prompts you to identify the purpose when you create the element.

Toolbox icon



Learn more

- [Sequence Diagram](#)^[1074]
- [Continuation](#)^[1155]
- [State Invariant](#)^[1157]

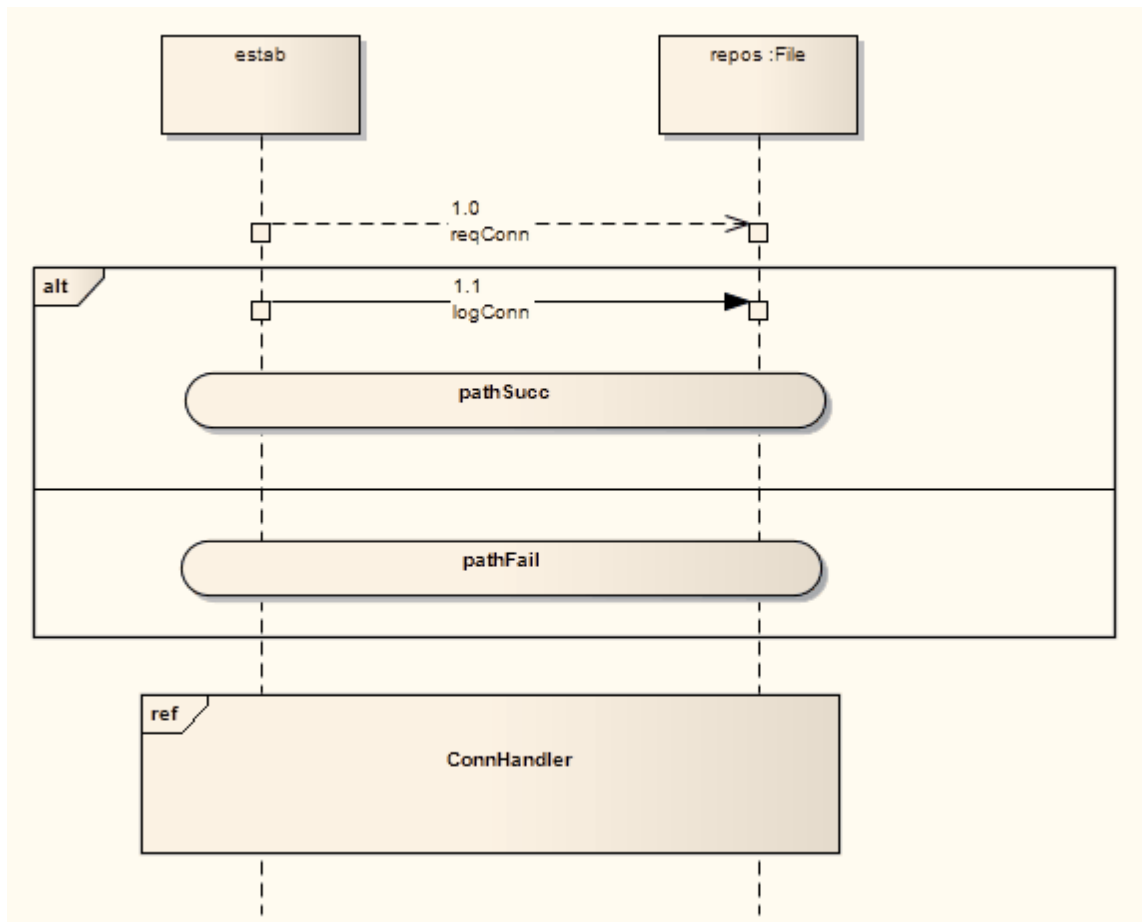
6.4.1.35.1 Continuation

Description

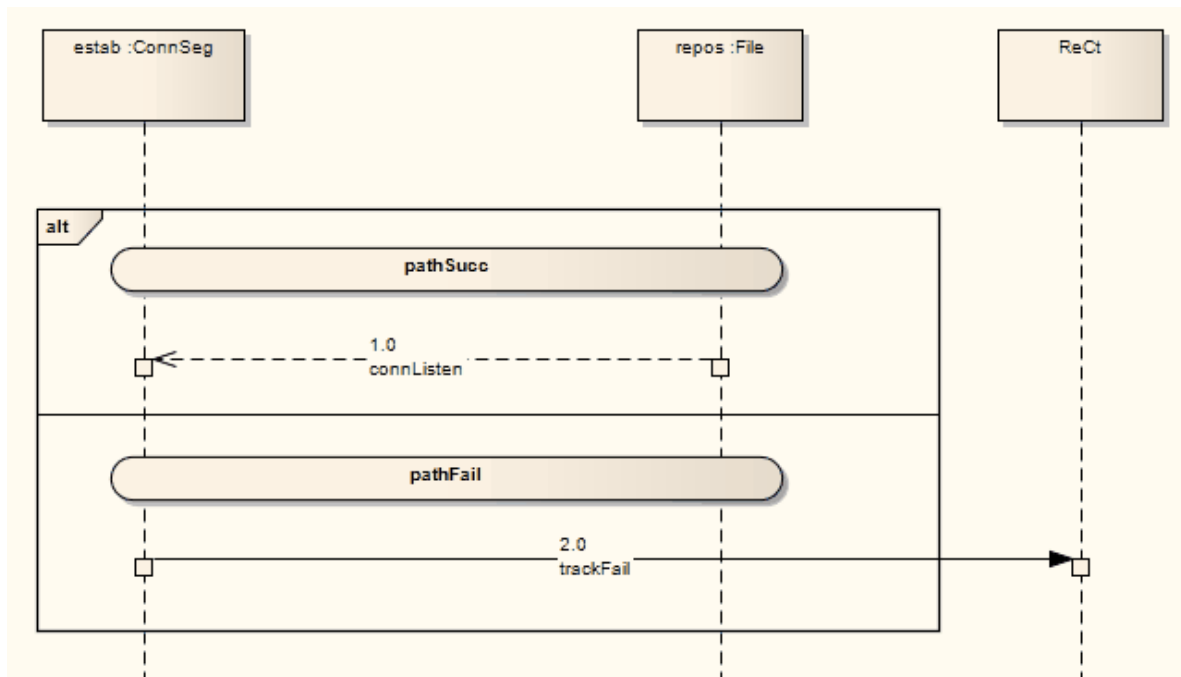
A Continuation is used in *seq* and *alt* Combined Fragments, to indicate the branches of continuation an operand follows. To indicate a continuation, end an operand with a Continuation, and indicate the continuation branch with a matching Continuation (same name) preceding the Interaction Fragment.

You create a Continuation by dragging the State/Continuation element onto the diagram from the Interaction Elements page of the Toolbox.

For the following continuation example, an *alt* Combined Fragment has Continuations *pathSucc* and *pathFail*. These Continuations are located within the Interaction Occurrence *ConnHandler*, which has subsequent events based on the continuation.



The following diagram shows the interaction referenced by the *Interaction Occurrence*.



Learn more

- [Combined Fragments](#) ^[1110]
- [State/Continuation Element](#) ^[1155]
- [Interaction Occurrence](#) ^[1139]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 474) states:

A Continuation is a syntactic way to define continuations of different branches of an Alternative CombinedFragment. Continuation is intuitively similar to labels representing intermediate points in a flow of control.

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 474) also states:

Continuations have semantics only in connection with Alternative CombinedFragments and (weak) sequencing.

If an InteractionOperand of an Alternative CombinedFragment ends in a Continuation with name (say) X, only InteractionFragments starting with the Continuation X (or no continuation at all) can be appended.

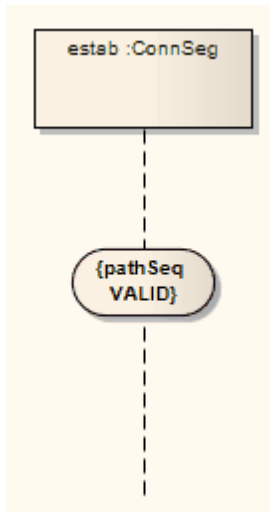
6.4.1.35.2 State Invariant

Description

A State Invariant is a condition applied to a Lifeline, which must be fulfilled for the Lifeline to exist. You create a State Invariant by dragging the State/Continuation element onto the diagram from the

Interaction Elements page of the Toolbox.

The following diagram illustrates a State Invariant.



When a State Invariant is moved near to a Lifeline, it snaps to the center. If the sequence object is dragged left or right, the State Invariant moves with it.

Learn more

- [Lifeline](#) ^[1144]
- [State/Continuation](#) ^[1155]

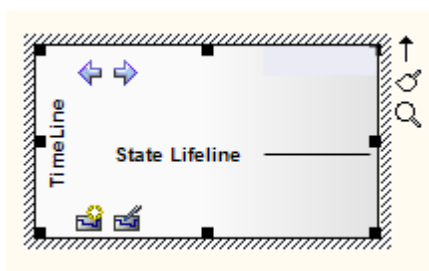
OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 502*) states:

A StateInvariant is a runtime constraint on the participants of the interaction. It may be used to specify a variety of different kinds of constraints, such as values of attributes or variables, internal or external states, and so on.

A StateInvariant is an InteractionFragment and it is placed on a Lifeline.

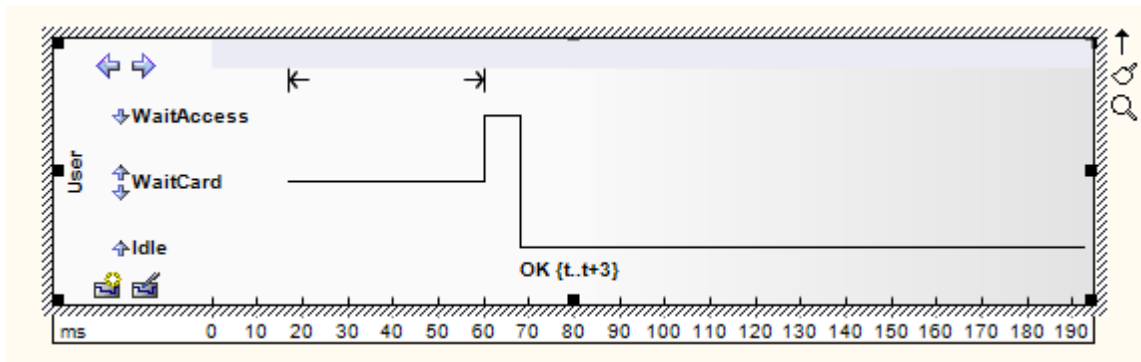
6.4.1.36 State Lifeline



Description

A Lifeline is the path an object takes across a measure of time, as indicated by the x-axis. There are two sorts: State Lifelines (defined here) and Value Lifelines, both used in Timing diagrams.

A State Lifeline follows discrete transitions between states, which are defined along the y-axis of the timeline. Any transition has optional attributes of timing constraints, duration constraints and observations. An example of a State Lifeline is shown below:



See *UML Superstructure Specification, v2.1.1, figure 14.29, p. 519.*

A State Lifeline consists of a set of transition points. Each transition point can be defined with the following properties:

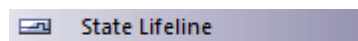
Property	Description
At time	Specifies the starting time for a change of state.
Transition to	Indicates the state to which the lifeline changes.
Event	Describes the occurring event.
Timing constraints	Refers to the time taken for a state to change within a lifeline, or the time taken to transmit a message (e.g. $t..t+3$).
Timing observations	Provides information on the time of a state change or sent message.
Duration constraints	Pertains to a lifeline's period at a particular state. The constraint could be instigated by a change of state within a lifeline, or that lifeline's receipt of a message.
Duration observations	Indicates the interval of a lifeline at a particular state, begun from a change in

Property	Description
	state or message receipt.

In the example diagram above, the **OK** transition point has these properties:

Property	Value
At Time	18 ms
Transition to	Idle
Event	OK
Timing constraints	t..t+3
Timing observations	–
Duration constraints	–
Duration observations	–

Toolbox icon



Learn more

- [Timing Diagram](#)^[105]
- [Value Lifelines](#)^[117]

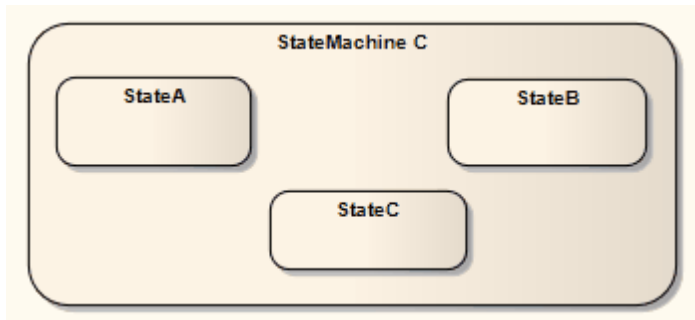
OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 518*) states:

This is the state of the classifier or attribute, or some testable condition, such as an discrete enumerable value.

It is also permissible to let the state-dimension be continuous as well as discrete. This is illustrative for scenarios where certain entities undergo continuous state changes, such as temperature or density.

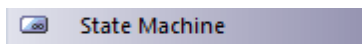
6.4.1.37 State Machine



Description

A State Machine element is a container for groups of related State elements. You can create sections of a State Machine diagram, showing the organization of the inter-related State elements, and enclose each section in a State Machine element. You can also create Regions on a State Machine element.

Toolbox icon



Learn more

- [Regions](#) ^[1037]

6.4.1.38 Structured Activity

Description

Structured Activity elements are used in Activity diagrams. A Structured Activity is an activity node that can have subordinate nodes as an independent *Activity Group*. You can set an option to ensure that no other Activities or their side effects interfere with this Activity's processing.

Enterprise Architect provides a number of forms of Structured Activity, both basic and specialized, for creating composite Activity elements quickly and simply.

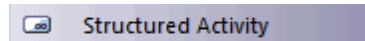
Access [View](#) | [Diagram Toolbox](#) | [More tools...](#) | [UML](#) | [Activity](#)

Topic	Description	See Also
Create Structured Activities	<p>When you drag a Structured Activity icon from the Toolbox onto a diagram, a short menu displays from which you select one of the following options:</p> <ul style="list-style-type: none"> • Loop Node • Conditional Node • Other 	<p>Loop Node ^[1164]</p> <p>Conditional Node ^[1165]</p>

Topic	Description	See Also
	<p>The first two options specifically create a Loop Node or Conditional Node.</p> <p>The Other option displays the New Structured Activity dialog, on which you can select to create one of the five following types of Structured Activity element.</p>	
Simple Composite Activity	Provides a Composite Activity element with a child Activity diagram.	Activity ^[1101] Composite Elements ^[825]
Loop Node	Represents a sequence of Actions and Activities that are to be repeated within the object.	Loop Node ^[1164]
Conditional Node	Represents an arrangement of Actions and Activities where choice determines which Activities are performed.	Conditional Node ^[1168]
Structured Activity Node	Represents an ordered arrangement of executable Activity nodes (Actions, Decisions, Merges and so on) that can include branched and nested nodes; this is the base element from which the other types of Structured Activity are derived.	Structured Node ^[1163]
Sequential Node	Represents a sequential arrangement of executable Activity nodes.	Sequential Node ^[1164]
Nodes as Composite Elements	<p>Loop Node and Conditional Node elements are composite elements, but the child element structures are created on the <i>same</i> diagram as the node element itself (in the same way as for a Composite State). Therefore:</p> <ul style="list-style-type: none"> • The composite element symbol is not set • You cannot display a separate child diagram • The Show Composite Diagram facility is not available for these nodes <p>You could set these nodes as 'normal' composite elements and create and open a child diagram, but you should not use Loop or Conditional Nodes in this way.</p> <p>The Show Composite Diagram facility is also not available on the Structured Activity Node, as this is the base element for the Loop and Conditional Nodes. You can, however:</p> <ul style="list-style-type: none"> • Use the Sequential and Structured Activity nodes as composite elements, and • Display a child diagram structure on its parent Sequential node 	Composite State ^[1153]

Topic	Description	See Also

Toolbox icon



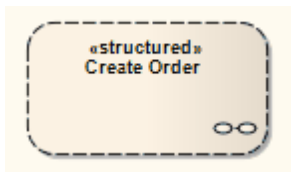
Learn more

- [Activity Diagram](#)^[1028]
- [Activity](#)^[1101]

6.4.1.38.1 Structured Node

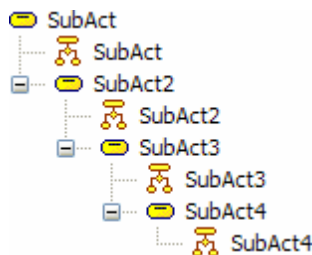
Description

On a diagram, *Structured Activity Nodes* have broken borders and composite diagram icons, as shown below:



To display the Activity diagram represented by a Structured Activity Node element, double-click on the element.

Structured Activity Node elements can point to child diagrams that themselves contain or consist of Structured Activity elements; that is, the Structured Activity elements are nested, as shown in the section of Project Browser below.



OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 409*) states:

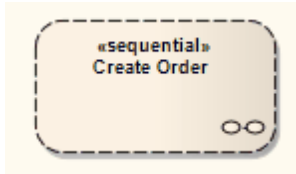
A structured activity node is an executable activity node that may have an expansion into subordinate nodes as an ActivityGroup. The subordinate nodes must belong to only one structured activity node, although they may be nested.

A structured activity node represents a structured portion of the activity that is not shared with any other structured node, except for nesting.

6.4.1.38.2 Sequential Node

Description

On a diagram, *Sequential* Activity Nodes have broken borders and composite diagram icons, as shown below:



To display the Activity diagram represented by a Sequential Activity Node element, double-click on the element.

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 408*) states:

A sequence node is a structured activity node that executes its actions in order.

6.4.1.38.3 Loop Node

Description

A *Loop* Structured Activity Node is used for defining a loop, and is commonly associated with 'While', 'Repeat' or 'For' loop statements.

A Loop Node has three partitions:

- **Setup** commonly initiates variables to be used in the loop's exit-condition; it is executed once on entry to the loop
- **Test** defines the loop exit-condition
- **Body** can contain Actions to be executed repeatedly until the Test produces a false value

The results of the final execution of the Test or Body are available after execution of the Loop is complete.

How to

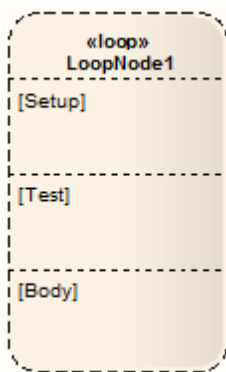
To create a Loop Node

Step	Action	See Also
1	From the Activity page of the Diagram Toolbox, drag a Structured Activity icon onto the Activity diagram.	

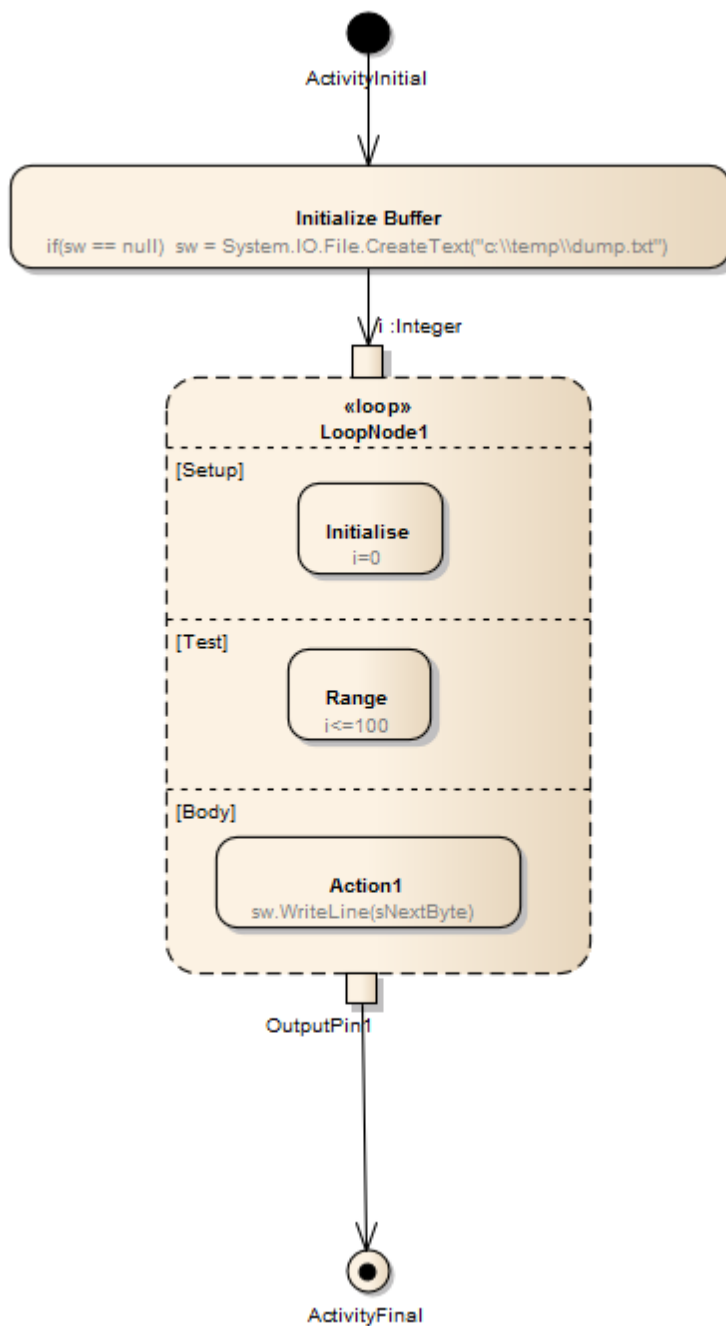
Step	Action	See Also
	A short menu displays.	
2	<p>Select the Loop Node option.</p> <p>The Loop Node displays on the diagram, with the element Properties dialog (if the dialog does not display, double-click on the element).</p>	
3	Complete as many of the common element Properties fields as required.	
4	<p>Select the Loop page and set the following checkboxes as required:</p> <ul style="list-style-type: none"> • Must Isolate - defines concurrency: if selected, no object within the node can be used outside it; the objects are isolated from parallel usage • Tested First - defines the loop type; select for a <i>For / While</i> loop, deselect for a <i>Repeat Until</i> loop 	
5	<p>For each of the following fields, click on the (...) or Add button as appropriate, to display the Select Pins dialog and select an Action Pin:</p> <ul style="list-style-type: none"> • Decider (an Output pin within the Test partition, the value of which is examined after execution of the Test to determine whether to execute the loop Body) • Loop Variable Input • Loop Variable • Body Output and • Result <p>The Select Pins dialog lists only Input pins for the Loop Variable Input field and only Output pins for the other fields.</p> <p>If the required Action Pin does not already exist, you can click on the Add New button on the dialog to automatically create the Input pin or an Output pin for the node.</p>	
6	<p>In the Nodes panel, click on one of the Setup, Test or Body radio buttons to list the Actions and Activities contained in the corresponding partition of the Loop Node.</p> <p>An element must be completely below the top edge of a partition to be listed for that partition - if it overlaps with the partition above in any way, it is treated as being part of that partition.</p>	
7	Click on the OK button to save the properties of the Loop Node and close the Properties dialog.	
8	Right-click on the Node in the diagram and select the Embedded Elements Embedded Elements context menu option.	

Step	Action	See Also
	<p>The Embedded Elements dialog displays.</p> <p>Select the checkbox against each embedded element and close the dialog.</p> <p>The Action pins should now be visible in the diagram, attached to the Node.</p>	

A Loop Node is depicted on an Activity diagram as shown below:



You define the Loop nodes by dragging Action elements from the Toolbox page into the Setup, Test and Body partitions. The Body partition can contain several Actions, which can be linked and organized into the required structure. The elements are aligned on the top left of the partition, so that resizing the node maintains the organization of the structure within and between the partitions. If you try to shrink the node below the structure size, the node automatically defaults to the 'best fit' size.



Notes

- You can check on the exact location of an existing Action Pin by right-clicking on the pin name in the Loop Node Properties dialog and selecting the **Find in Project Browser** context menu option; the location of the Action Pin in the Project Browser is expanded and highlighted

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, pp. 384-385*) states:

A loop node is a structured activity node that represents a loop with setup, test, and body sections.

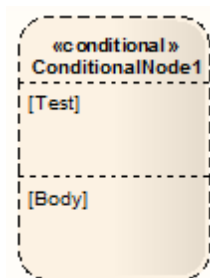
Each section is a well-nested subregion of the activity whose nodes follow any predecessors of the loop and precede any successors of the loop. The test section may precede or follow the body section. The setup section is executed once on entry to the loop, and the test and body sections are executed repeatedly until the test produces a false value. The results of the final execution of the test or body are available after completion of execution of the loop.

6.4.1.38.4 Conditional Node

Description

A Conditional Structured Activity Node has a default *Clause* containing *Test* and *Body* partitions. Further Clauses can be added if required.

They are depicted on an Activity diagram as shown below:



You define Conditional nodes by dragging other Activity diagram elements from the Toolbox page into the appropriate partition of the element, and linking and organizing the structure as required. The elements are aligned on the top left of the partition, so that resizing the node maintains the organization of the structure within and between the partitions. If you try to shrink the node below the structure size, the node automatically defaults to the 'best fit' size.

When you create a Conditional Node, the element Properties dialog displays. Much of this you can complete as for any other element. However, for the Conditional Node the dialog shows an additional page.

Add an Action Pin as the **Result** for the node, clicking on the **Add** button to display the Select Pins dialog (a version of the Select <Item> dialog).

On creation, the Conditional Node automatically has one *Clause* containing a **Decider** and **Body Output**, and a Test partition and a Body partition. You can add further Clauses as required. For each Clause you also add an Action Pin for the **Decider** and for the **Body Output**. Click on the **Save** button to save the Clause definition.

The Select Pin dialog reveals only Output pins as appropriate to the context. If the required Action Pin does not already exist, you can click on the **Add New** button on the dialog to automatically create an Output pin under the appropriate parent node.

For the **Result** and **Body Output** entries, you can check on the exact location of each Action Pin by right-clicking on the entry and selecting the **Find in Project Browser** context menu option.

The Nodes panel, by default, lists the Actions and Activities contained in the Test partition. Click on the **Body** radio button to list the elements contained in the Body partition. An element must be completely contained in the Body partition to be listed there - if it overlaps with the Test partition in any way, it is treated as being part of the Test partition.

Add or Remove Clauses:

To add another Clause, click on the **Add** button underneath the Clause(s) list. This inserts a new Clause in the list, and identifies which is the preceding, or Predecessor, Clause and (if appropriate) which is the following, or Successor, Clause. The remaining fields in the Clause(s) panel are cleared to enable you to add **Decider** and **Body Output** Action Pins. New Test and Body partitions are immediately added to the element on the diagram, and you can populate these partitions with Activity elements, which are then identified in the Nodes panel.

To remove a Clause, highlight it in the list and click on the **Delete** button. This immediately removes the Clause's corresponding partitions from the diagram, along with all their contained Activity elements. Removing a Clause from between two other Clauses adjusts the numerical order; for example, if Clause 2 is removed from between Clause 1 and Clause 3, Clause 3 is renamed as Clause 2, and any further Clauses are also moved up one place.

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p.355*) states:

A conditional node is a structured activity node that represents an exclusive choice among some number of alternatives.

A conditional node consists of one or more clauses. Each clause consists of a test section and a body section. When the conditional node begins execution, the test sections of the clauses are executed. If one or more test sections yield a true value, one of the corresponding body sections will be executed. If more than one test section yields a true value, only one body section will be executed. The choice is nondeterministic unless the test sequence of clauses is specified. If no test section yields a true value, then no body section is executed; this may be a semantic error if output values are expected from the conditional node.

6.4.1.39 Synch



Description

A Synch state is useful for indicating that concurrent paths of a State Machine are synchronized. After bringing the paths to a synch state, the emerging transition indicates unison.

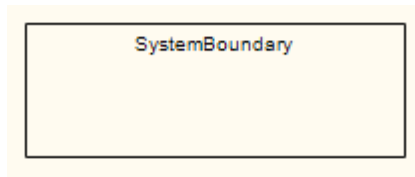
Toolbox icon



Learn more

- [State Machine Diagram](#) ^[1032]

6.4.1.40 System Boundary



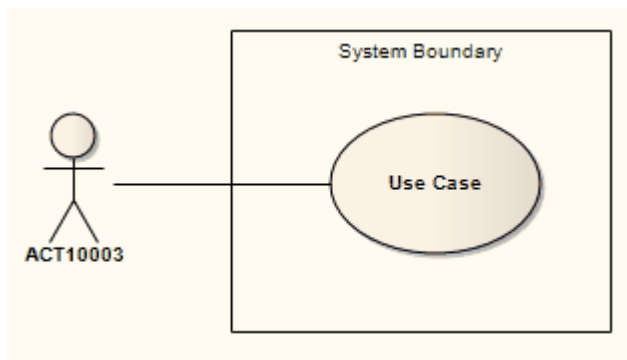
Description

A *System Boundary* element is a non-UML element used to define *conceptual* boundaries. You can use System Boundaries to help group logically related elements (from a visual perspective, not as part of the UML model).

In the *UML Superstructure Specification, v2.1.1*, System Boundaries are described in the sections on Use Cases, because the System Boundary is often used to indicate the *application* of a Use Case to another entity. In this context, the System Boundary:

- Encloses the Use Case, and
- Is associated with a classifier such as a Class, Component or Sub-system (Actor) through the Select <Item> dialog

By associating the System Boundary - and not the Use Case - with the classifier, the classifier is linked to the Use Case as a *user*, but not as an *owner*.



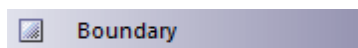
You can also define a Use Case as the classifier of a System Boundary element, to link the elements enclosed in the System Boundary (such as parts of an Activity diagram) to their representation in a logical Use Case. See http://www.sparxsystems.com.au/resources/map_uc.html.

The following properties of a System Boundary can be set: the name, the border style, and the number of horizontal or vertical swim lanes.

A System Boundary element can be marked as *Selectable*, using the element's context menu. When not selectable, you can click within the System Boundary space without activating or selecting the Boundary itself. This is useful when you have many elements within the Boundary and the Boundary makes their selection difficult.

Notes

- A System Boundary is the basis for the Image element, which enables you to add icons or backgrounds to a diagram, automatically displaying the Image Manager dialog from which to select the appropriate image

Toolbox icon**Learn more**

- [Use Cases](#) ^[1174]
- [Instance Classifier](#) ^[880]
- [Boundary Element Settings](#) ^[1171]

OMG UML Specification:

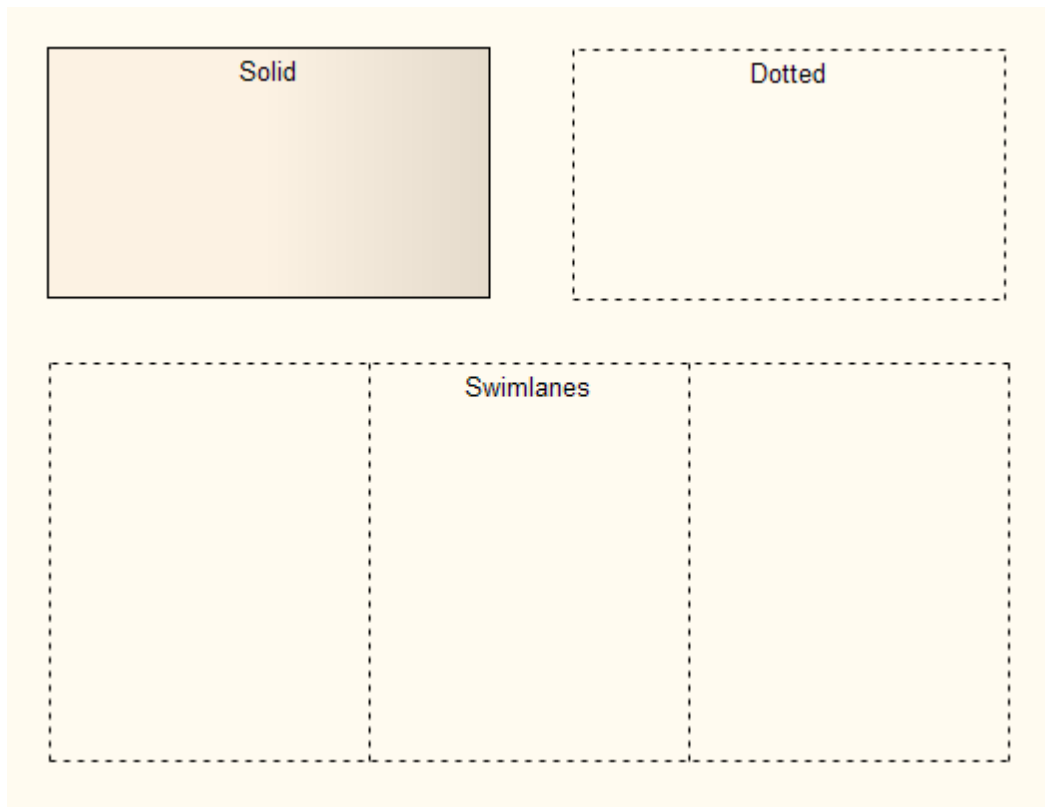
The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 594*) states:

If a subject (or system boundary) is displayed, the Use Case ellipse is visually located inside the system boundary rectangle. Note that this does not necessarily mean that the subject classifier owns the contained Use Cases, but merely that the Use Case applies to that classifier.

6.4.1.40.1 Boundary Element Settings**Configure Boundary Elements:**

Boundary elements can be configured to display in different ways. The main differences are:

- Solid border
- Dotted border
- With horizontal or vertical 'swim lanes'; swim lanes are used to group elements in a vertical or horizontal context (for example, Client, Application and Database tiers could be represented in swim lanes)



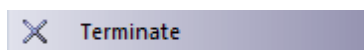
6.4.1.41 Terminate



Description

The Terminate pseudo-state indicates that upon entry of its pseudo-state, the State Machine's execution ends.

Toolbox icon



Learn more

- [State Machine Diagram](#)^[1032]
- [Pseudo-state](#)^[1037]

6.4.1.42 Trigger



Description

A Trigger indicates an event that initiates an action (and might arise from completion of a previous action). You initially define a Trigger in one of four ways:

- As a property of a Transition relationship
- As a property of an Accept Event Action (on the Triggers tab of the element Properties dialog)
- As an event in a State Machine Table
- Directly, as a Trigger element, through the New Element dialog

When you save the Trigger, it is added to the list of elements for the parent package in the Project Browser. You can then right-click on it and select the **Properties** context menu option to view and, if required, edit its properties as an element rather than as a property itself. Triggers created as events remain as Event elements, whilst Triggers created in other ways are Trigger elements, with a Trigger tab in the Properties dialog.

Field	Action	See also
Type	<p>If necessary, edit the type of trigger:</p> <ul style="list-style-type: none"> • Call - specifies that the event is a CallEvent, which sends a message to the associated object by invoking an operation • Change - specifies that the event is a ChangeEvent, which indicates that the transition is the result of a change in value of an attribute • Signal - specifies that the event is a SignalEvent, which corresponds to the receipt of an asynchronous signal instance • Time - corresponds to a TimeEvent; which specifies a moment in time 	
Specification	<p>Either type in the event instigating the Trigger, or click on the (...) button and select the event (depending on the Type value).</p>	
Ports	<p>Click on the Add button and select the appropriate Port from the Select Port dialog.</p> <ul style="list-style-type: none"> • To create new Ports using the Select Port dialog, the 	Select Port Dialog <small>1880</small>

Field	Action	See also
	<p>Trigger should be created as a child of a Class or Component element</p> <ul style="list-style-type: none"> To add several Ports at once, press (Ctrl) as you select each Port To check the exact location of a Port, right-click on the Port name and select the Find in Project Browser context menu option 	

You can also drag the Trigger element onto another diagram, although there are limited uses for the element in that context.

This element is not the same as a Trigger Operation, which is an operation automatically executed as a result of the modification of data in a database.

Learn more

- [Action](#)^[109]
- [Transition Relationship](#)^[126]
- [Insert Trigger](#)^[104]
- [Trigger Operation](#)^[196]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 456*) states:

Events may cause execution of behavior (e.g., the execution of the effect activity of a transition in a state machine). A trigger specifies the event that may trigger a behavior execution as well as any constraints on the event to filter out events not of interest.

6.4.1.43 Use Case

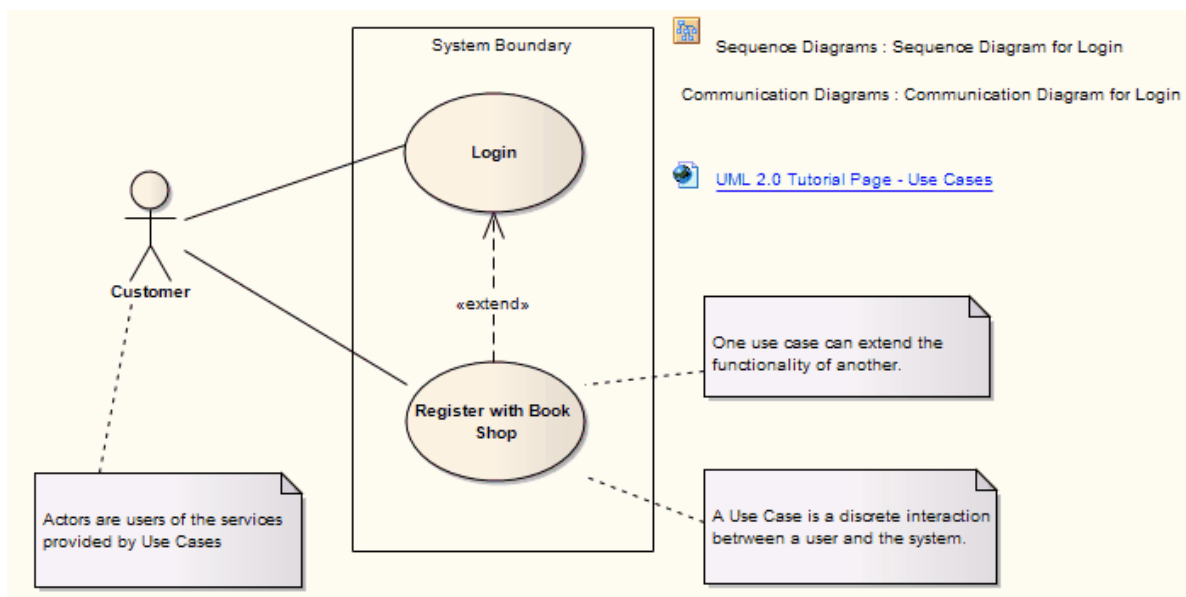


Description

A Use Case is a UML modeling element that describes how a user of the proposed system interacts with the system to perform a discrete unit of work. It describes and signifies a single interaction over time that has meaning for the end user (person, machine or other system), and is required to leave the system in a complete state: the interaction either completed or rolled back to the initial state. A Use Case:

- Typically has requirements and constraints that describe the essential features and rules under which it operates
- Can have an associated Sequence diagram illustrating behavior over time; who does what to whom, and when
- Typically has scenarios associated with it that describe the work flow over time that produces the end result; alternative work flows (for example, to capture exceptions) are also enabled

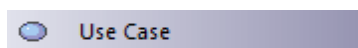
The following is an example Use Case diagram:



If extending a Use Case, you can specify the points of extension with Use Case Extension Points. To display the attributes, operations or constraints of a Use Case on a diagram, use Rectangle Notation.

Enterprise Architect also provides two stereotyped Use Cases: the Test Case and the Business Use Case.

Toolbox icon



Learn more

- [Sequence Diagram](#)^[1074]
- [Use Case Extension Points](#)^[1176]
- [Rectangle Notation](#)^[1177]
- [Test Case](#)^[1677]
- [Business Use Case](#)^[1514]

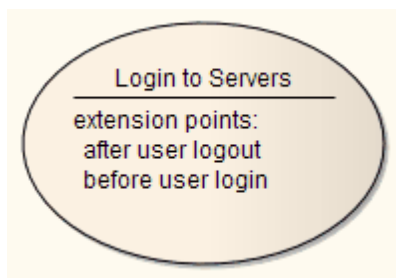
OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 592*) states:

A UseCase is a kind of behavioered classifier that represents a declaration of an offered behavior. Each Use Case specifies some behavior, possibly including variants, that the subject can perform in collaboration with one or more actors.

6.4.1.43.1 Use Case Extension Points

Use *extension points* to specify the point of an extended Use Case where an extending Use Case's behavior should be inserted. The specification text can be informal or precise to define the location of the extension point.

**How to**

To work with extension points

Step	Action
1	Right-click on the Use Case element, and select the Advanced Edit Extension Points... context menu option. The Use Case Extension Points dialog displays, listing defined points for that Use Case.
2	Select an extension point in the list and click on the appropriate button to edit or remove the extension point, or to add a new one.

Notes

- Conditions to apply the extending Use Case, and the extension point to use, should be attached as a note to the extend relationship

Learn more

- [Use Case](#)^[1174]

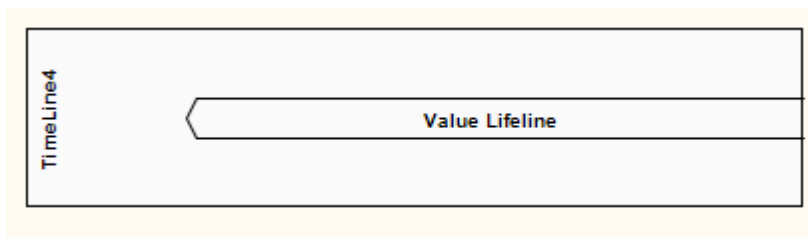
6.4.1.43.2 Rectangle Notation

You can display various shaped elements, such as an Interface, Use Case or Actor, using *rectangle notation*. This displays the element as a rectangle, with an icon of the 'normal' shape in the top right-hand corner. Any attributes, operations or constraints belonging to the element are shown, in the same style as a Class.



To show an element using rectangle notation, right-click on the element on the diagram and select the **Advanced | Use Rectangle Notation** context menu option. This setting only applies to the selected element, and can be toggled on and off either by deselecting the context menu option or by selecting the reciprocal option such as **Use Circle Notation** or **Use Actor Notation**.

6.4.1.44 Value Lifeline

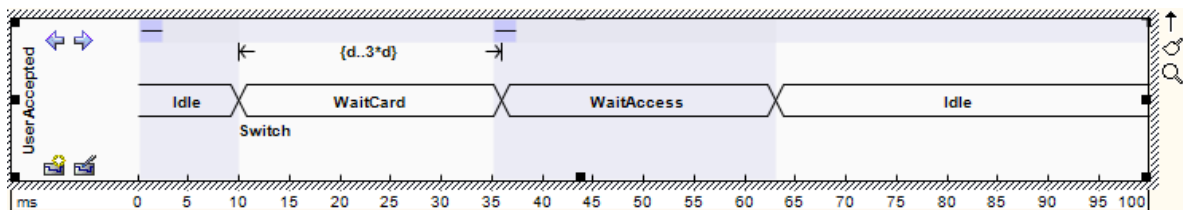


Description

A Lifeline is the path an object takes across a measure of time, indicated by the x-axis. There are two sorts: Value Lifelines (defined here) and State Lifelines, both used in Timing diagrams.

A Value Lifeline shows the Lifeline's state across the diagram, with parallel lines indicating a steady state. A cross between the lines indicates a transition or change in state.

An example of a Value Lifeline is shown below:



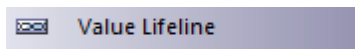
See *UML Superstructure Specification, v2.1.1, Figure 14.30, p. 520*.

A Value Lifeline consists of a set of transition points. Each transition point can be defined with the following properties:

Property	Description
At time	Specifies the starting time for a change of state.
Transition to	Indicates the state to which the Lifeline is to change.
Event	Describes the occurring event.
Timing constraints	Refers to the time taken for a state to change within a Lifeline, or the time taken to transmit a message.
Timing observations	Provides information on the time of a state change or sent message.
Duration constraints	Pertains to a Lifeline's period at a particular state. The constraint could be instigated by a change of state within a Lifeline, or that Lifeline's receipt of a message.
Duration observations	Indicates the interval of a Lifeline at a particular state, begun from a change in state or message receipt.

In the example diagram above, the **10ms** transition point has these properties:

Property	Text
At Time	10ms
Transition to	Waitcard
Event	Switch
Timing constraints	–
Timing observations	–
Duration constraints	d..3*d
Duration observations	–

Toolbox icon**Learn more**

- [Timing Diagrams](#) ^[1051]
- [State Lifelines](#) ^[1158]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 518*) states:

Shows the value of the connectable element as a function of time. Value is explicitly denoted as text. Crossing reflects the event where the value changed.

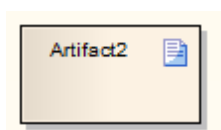
6.4.2 Structural Diagram Elements

The following elements are used in UML Structural Diagrams. For more information on using each element, click on the element name in this list:

- [Actor](#) ^[1107], [Artifact](#) ^[1179]
- [Class](#) ^[1181], [Collaboration](#) ^[1184], [Collaboration Use](#) ^[1186], [Component](#) ^[1187]
- [Data Type](#) ^[1188], [Deployment Specification](#) ^[1189], [Document Artifact](#) ^[1191]
- [Enumeration](#) ^[1191], [Execution Environment](#) ^[1192], [Expose Interface](#) ^[1193]
- [Information Item](#) ^[1194], [Interface](#) ^[1194]
- [Node](#) ^[1196], [Note](#) ^[1147]
- [Object](#) ^[1196]
- [Package](#) ^[1200], [Part](#) ^[1200], [Port](#) ^[1202], [Primitive](#) ^[1204]
- [Qualifiers](#) ^[1211]
- [Signal](#) ^[1205]

Learn more

- [Structural Diagrams](#) ^[1011]

6.4.2.1 Artifact**Description**

An *Artifact* is any physical piece of information used or produced by a system, represented in a Deployment diagram.

Artifacts can have associated properties or operations, and can be instantiated or associated with other Artifacts. Examples of Artifacts include model files, source files, database tables, development deliverables or support documents. The files represented by the Artifact are listed on the Files tab of the element Properties dialog.

To open the files represented by the Artifact, click on the element on the diagram and press **(Ctrl+E)**. Each file is opened either on a separate tab in the Diagram View workspace (if the file can be opened within Enterprise Architect) or in the default Windows viewer/editor for the file type (if the file cannot be opened within Enterprise Architect).

Files can also be launched individually from the Files tab (opening in the Windows default editor), as for elements of any other type that have associated files.

Toolbox icon



Learn more

- [Create Artifact For External File](#) ^[1180]
- [Deployment Diagram](#) ^[1020]
- [Associated Files](#) ^[875]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 201*) states:

An Artifact defined by the user represents a concrete element in the physical world. A particular instance (or 'copy') of an artifact is deployed to a node instance. Artifacts may have composition associations to other artifacts that are nested within it. For instance, a deployment descriptor artifact for a component may be contained within the artifact that implements that component. In that way, the component and its descriptor are deployed to a node instance as one artifact instance.

6.4.2.1.1 Create Artifact For External File

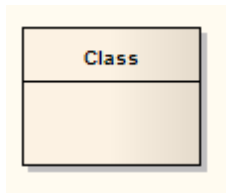
You can create an Artifact element on a diagram for an external file, by clicking on the file in a file list (such as Windows Explorer) or on your Desktop and dragging it onto the diagram. A short context menu displays with two options - **Hyperlink** and **Artifact**.

Click on the **Artifact** option to create the element on the diagram.

The element Properties dialog displays, and you can enter the element's name and other properties as required.

Click on the **OK** button, and then open the Properties dialog again and click on the Files tab. The file pathname is listed in the Files panel.

6.4.2.2 Class



Description

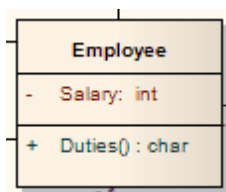
A Class is a representation of a type of object that reflects the structure and behavior of such objects within the system. It is a template from which actual running instances are created, although a Class can be defined either to control its own execution or as a template or parameterized Class that specifies parameters that must be defined by any binding Class.

A Class can have *attributes* (data) and *methods* (operations or behavior). Classes can inherit characteristics from parent Classes and delegate behavior to other Classes. Class models usually describe the logical structure of the system and are the building blocks from which components are built.

The top section of a Class, as illustrated below, shows the attributes (or data elements) associated with the Class. These hold the 'state' of an object at run-time. If the information is saved to a data store and can be reloaded, it is termed 'persistent'. The lower section contains the Class operations (or methods at run-time). Operations describe the behavior a Class offers to other Classes, and the internal behavior it has (private methods).

Class elements are generally used in Class diagrams and Composite Structure diagrams.

Enterprise Architect also supports a number of stereotyped Class elements to represent various entities in web-page modeling. A Class can also be integrated with an Associate connector to form an Association Class, to allow the Associate connector to have operations and attributes that define certain types of UML relationship.



Toolbox icon



Learn more

- [Active Classes](#) ^[1182]
- [Parameterized Classes \(Templates\)](#) ^[1183]
- [Class Diagrams](#) ^[1014]
- [Composite Structure Diagram](#) ^[1017]
- [Association Class](#) ^[1215]

- [Association](#)^[1210]
- [Attributes](#)^[885]
- [Operations](#)^[900]
- [Web-page Modeling](#)^[1660]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, pp. 52-53*) states:

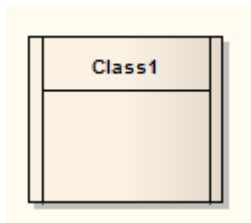
The purpose of a class is to specify a classification of objects and to specify the features that characterize the structure and behavior of those objects.

Objects of a class must contain values for each attribute that is a member of that class, in accordance with the characteristics of the attribute, for example its type and multiplicity.

When an object is instantiated in a class, for every attribute of the class that has a specified default, if an initial value of the attribute is not specified explicitly for the instantiation, then the default value specification is evaluated to set the initial value of the attribute for the object.

Operations of a class can be invoked on an object, given a particular set of substitutions for the parameters of the operation. An operation invocation may cause changes to the values of the attributes of that object. It may also return a value as a result, where a result type for the operation has been defined. Operation invocations may also cause changes in value to the attributes of other objects that can be navigated to, directly or indirectly, from the object on which the operation is invoked, to its output parameters, to objects navigable from its parameters, or to other objects in the scope of the operation's execution. Operation invocations may also cause the creation and deletion of objects.

6.4.2.2.1 Active Classes



Description

An *Active Class* indicates that, when instantiated, the Class controls its own execution. Rather than being invoked or activated by other objects, it can operate standalone and define its own thread of behavior.

How to

To define an Active Class in Enterprise Architect

Step	Action
1	Highlight a Class, and display its Properties dialog.
2	Click on the Advanced button.

Step	Action
3	Select the Is Active checkbox.
4	Click on the OK button to save the changes.

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 438*) states:

An active object is an object that, as a direct consequence of its creation, commences to execute its classifier behavior, and does not cease until either the complete behavior is executed or the object is terminated by some external object. (This is sometimes referred to as "the object having its own thread of control.") The points at which an active object responds to communications from other objects is determined solely by the behavior of the active object and not by the invoking object. If the classifier behavior of an active object completes, the object is terminated.

6.4.2.2 Parameterized Classes (Templates)

Description

Enterprise Architect supports *template* or *parameterized Classes*, which specify parameters that must be defined by any binding Class.

Parameterized Classes are commonly implemented in C++; Enterprise Architect imports and generates templated Classes for C++.

A template Class enables its functionality to be reused by any bound Class. If a default value is specified for a parameter, and a binding Class doesn't provide a value for that parameter, the default is used.

How to

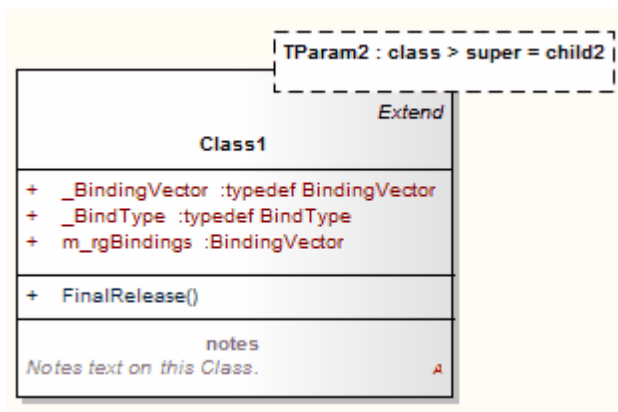
To create a parameterized Class

Step	Action
1	Display the Properties dialog for the required Class.
2	Select the Templates page.
3	In the Template Parameter(s) panel, click on the Add button. The Template Parameter dialog displays.
4	Type in the name and type of the parameter and, if required, click on the (...) buttons after the Constraints and Default fields to select the required constraining and default Classes from the Select <Item> dialog.

Step	Action
	The default Class can be either the constraining classifier or any Class that derives from the constraining classifier.

Notation Example:

On a diagram, template Classes are shown with the parameters in a dashed outline box in the upper right corner of the Class.



OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 622*) states:

A template is a parameterized element that can be used to generate other model elements using TemplateBinding relationships. The template parameters for the template signature specify the formal parameters that will be substituted by actual parameters (or the default) in a binding.

6.4.2.3 Collaboration

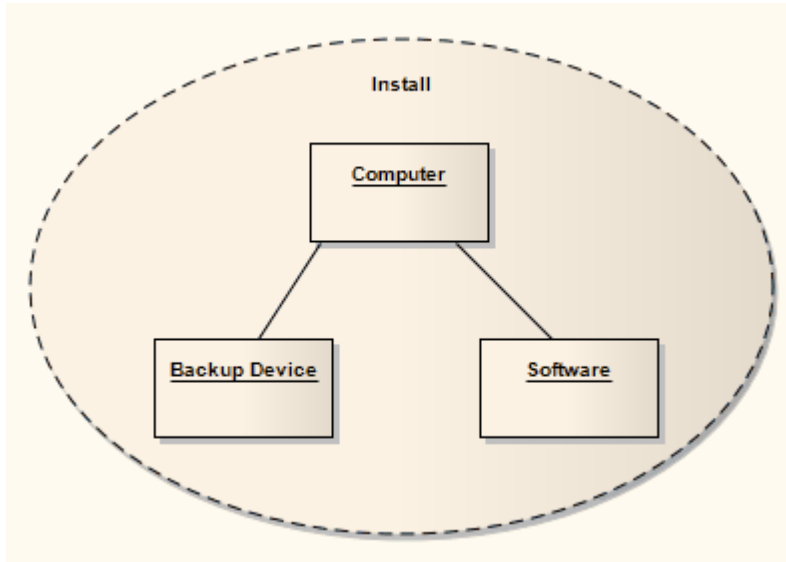


Description

A *Collaboration* defines a set of cooperating roles and their connectors. These are used to collectively illustrate a specific functionality, in a Composite Structure diagram. A Collaboration should specify only the roles and attributes required to accomplish a specific task or function. Although in practice a behavior and its roles could involve many tangential attributes and properties, isolating the primary roles and their requisites simplifies and clarifies the behavior, as well as providing for reuse. A Collaboration often implements a pattern to apply to various situations.

The following example illustrates an *Install* Collaboration, with three roles (Objects) connected as shown. The process for this Collaboration can be demonstrated by attaching an Interaction diagram (Sequence,

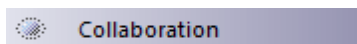
Timing, Communication or Interaction Overview).



To understand referencing a Collaboration in a specific situation, see the *Collaboration Use* topic.

Enterprise Architect supports a stereotyped Collaboration to represent a Business Use Case Realization in business modeling.

Toolbox icon



Learn more

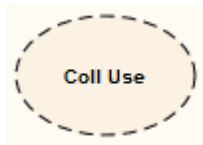
- [Composite Structure Diagram](#)^[1017]
- [Sequence Diagram](#)^[1074]
- [Timing Diagram](#)^[1057]
- [Communication Diagram](#)^[1084]
- [Interaction Overview Diagram](#)^[1087]
- [Collaboration Use](#)^[1186]
- [Object Element](#)^[1196]
- [Business Use Case Realization](#)^[1514]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 171*) states:

A collaboration describes a structure of collaborating elements (roles), each performing a specialized function, which collectively accomplish some desired functionality. Its primary purpose is to explain how a system works and, therefore, it typically only incorporates those aspects of reality that are deemed relevant to the explanation.

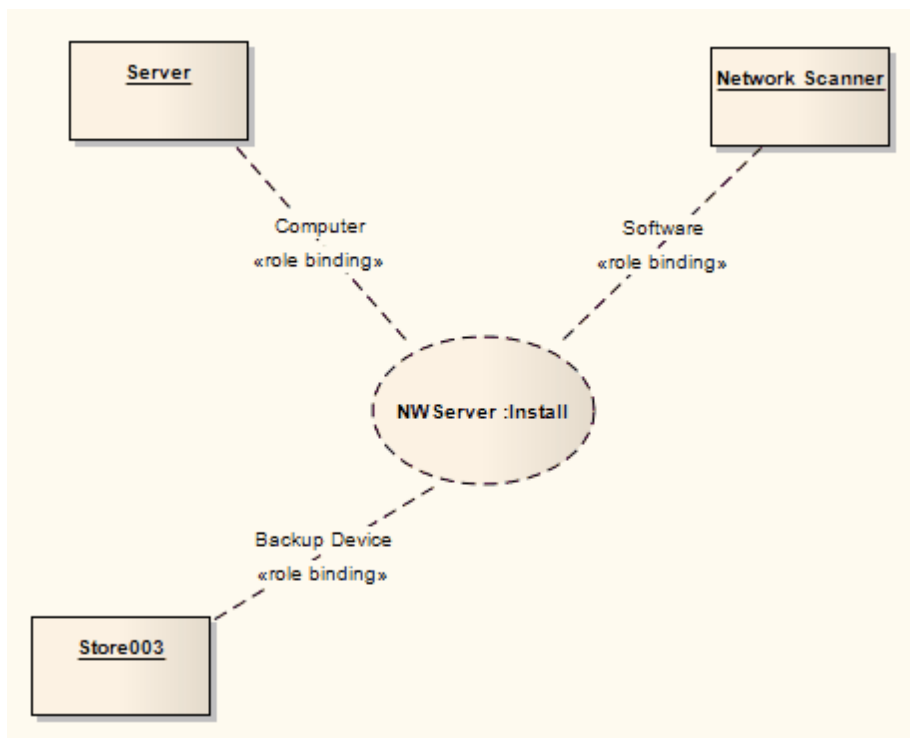
6.4.2.4 Collaboration Use



Description

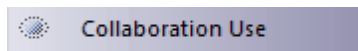
Use a *Collaboration Use* to apply a pattern defined by a Collaboration to a specific situation, in a Composite Structure diagram.

The following example shows a Use, *NWServer*, of the Collaboration *Install*, to define the installation process of a network scanner. This process can be defined by an interaction attached to the Collaboration. (See the *Collaboration* topic for a representation of the *Install* Collaboration.)



To create a Collaboration Use, drag the icon from the Toolbox onto the diagram.

Toolbox icon



Learn more

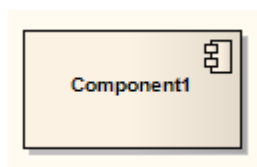
- [Composite Structure Diagram](#) ^[1017]
- [Collaboration Element](#) ^[1184]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 173*) states:

A collaboration use represents one particular use of a collaboration to explain the relationships between the properties of a classifier. A collaboration use shows how the pattern described by a collaboration is applied in a given context, by binding specific entities from that context to the roles of the collaboration. Depending on the context, these entities could be structural features of a classifier, instance specifications, or even roles in some containing collaboration. There may be multiple occurrences of a given collaboration within a classifier, each involving a different set of roles and connectors. A given role or connector may be involved in multiple occurrences of the same or different collaborations.

6.4.2.5 Component

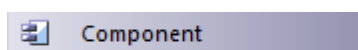


Description

A Component is a modular part of a system, whose behavior is defined by its provided and required interfaces; the internal workings of the Component should be invisible and its usage environment-independent. Source code files, DLLs, Java beans and other artifacts defining the system can be manifested in Components.

A Component can be composed of multiple Classes, or Components pieced together. As smaller Components come together to create bigger Components, the eventual system can be modeled, building-block style, in Component diagrams. By building the system in discrete Components, localization of data and behavior enables decreased dependency between Classes and Objects, providing a more robust and maintainable design.

Toolbox icon



Learn more

- [Component Diagram](#)^[1023]
- [Class Element](#)^[1187]
- [Object Element](#)^[1196]

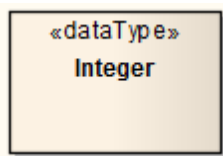
OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 148*) states:

A component represents a modular part of a system that encapsulates its contents and whose manifestation is replaceable within its environment.

A component defines its behavior in terms of provided and required interfaces. As such, a component serves as a type whose conformance is defined by these provided and required interfaces (encompassing both their static as well as dynamic semantics).

6.4.2.6 Data Type



Description

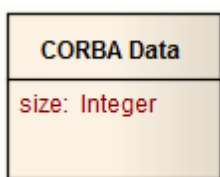
A Data Type is a specific kind of classifier, similar to a Class except that a Data Type cannot own sub Data Types, and instances of a Data Type are identified only by their value. For example, an instance of a Person Class is a Helen object, but an instance of an Integer Data Type is 12.

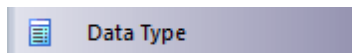
All copies of an instance of a Data Type, and any instances of that Data Type with the same value, are considered to be the same instance. That is, instances of Helen are not necessarily the same Helen, but all 12s are the same 12. For example, the 12 on a watch face is exactly the same integer as the number of months in a year.

Instances of a Data Type that have attributes (that is, are instances of a structured Data Type) are considered to be the same if the structure is the same and the values of the corresponding attributes are the same. If a Data Type has attributes, instances of that Data Type contain attribute values matching the attributes.

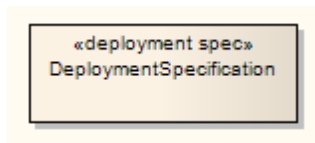
A typical use of Data Types would be to represent programming language primitive types or CORBA basic types. For example, integer and string types are often treated as Data Types.

A Data Type is denoted by a rectangle with the keyword «dataType», as above or, when it is referenced by (for example) an attribute, by a string containing the name of the Data Type, as below:



Toolbox iconLearn more

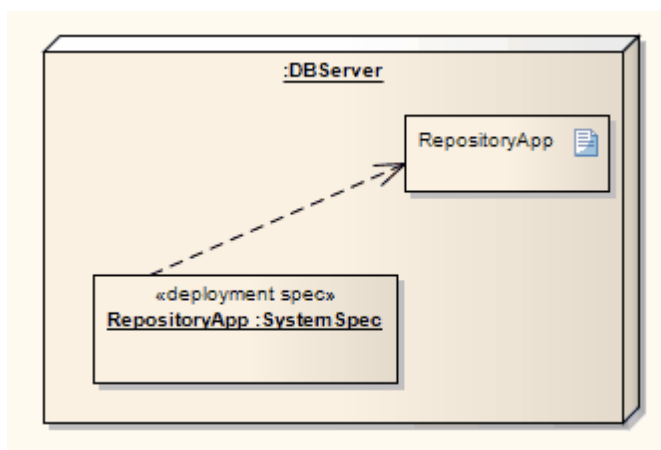
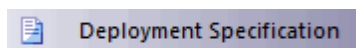
- [Class Element](#)^[1187]

6.4.2.7 Deployment SpecDescription

A Deployment Specification (spec) specifies parameters guiding deployment of an artifact, as is necessary with most hardware and software technologies. A specification lists those properties that must be defined for deployment to occur, as represented in a Deployment diagram. An instance of this specification specifies the values for the parameters; a single specification can be instantiated for multiple artifacts.

These specifications can be extended by certain component profiles. Examples of standard Tagged Values that a profile might add to a Deployment Specification are «*concurrencyMode*» with Tagged Values {*thread*, *process*, *none*} or «*transactionMode*» with Tagged Values {*transaction*, *nestedTransaction*, *none*}.

The following example depicts the artifact *RepositoryApp* deployed on the server node, as per the specifications of *RepositoryApp*, instantiated from the *Deployment Specification System Spec*.

Toolbox icon

Learn more

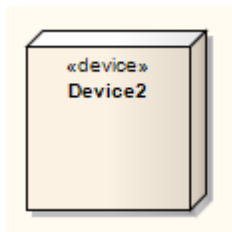
- [Deployment Diagram](#)^[1020]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 206*) states:

A deployment specification specifies a set of properties that determine execution parameters of a component artifact that is deployed on a node. A deployment specification can be aimed at a specific type of container. An artifact that reifies or implements deployment specification properties is a deployment descriptor.

6.4.2.8 Device



Description

A Device is a physical electronic resource with processing capability upon which Artifacts can be deployed for execution, as represented in a Deployment diagram. Complex Devices can consist of other devices; that is, a Device can be a nested element, where a physical machine is decomposed into its elements either through namespace ownership or through attributes that are typed by Devices.

Toolbox icon



Learn more

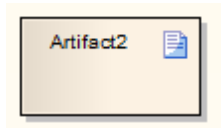
- [Deployment Diagram](#)^[1020]
- [Artifact Element](#)^[1179]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, 10.3.7, v2.1.1, p. 207*) states:

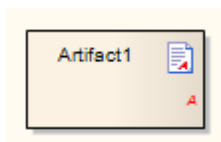
In the metamodel, a Device is a subclass of Node.

6.4.2.9 Document Artifact

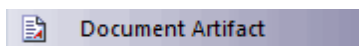


Description

A Document Artifact is an artifact having a stereotype of «*document*». You create the Document Artifact on a Component, Documentation or Deployment diagram, and associate it with an RTF document. Double-click on the element to display the Linked Document Editor. See the *Linked Documents* topic. When you have created the linked document, the Document Artifact element on the diagram shows an **A** symbol in the bottom right corner.



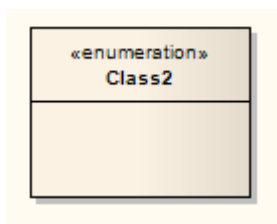
Toolbox icon



Learn more

- [Linked Documents](#)^[928]
- [Artifact Element](#)^[1179]


6.4.2.10 Enumeration



Description

An *Enumeration* is a data type, whose instances can be any of a number of user-defined enumeration literals. It is possible to extend the set of applicable enumeration literals in other packages or profiles. You create Enumerations in Class or Package diagrams, and in diagrams developed from the Metamodel and Profile pages of the **Toolbox**.

Toolbox icon

 EnumerationLearn more

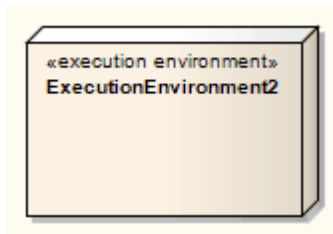
- [Class Diagram](#) ^[1014]
- [Package Diagram](#) ^[1017]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 69*) states:


An enumeration is a data type whose values are enumerated in the model as enumeration literals.

6.4.2.11 Execution Environment

Description

An Execution Environment is a node that offers an execution environment for specific types of components that are deployed on it in the form of executable artifacts. This is depicted in a Deployment diagram.

Execution Environments can be nested; for example, a database Execution Environment can be nested in an operating system Execution Environment. Components of the appropriate type are then deployed to specific Execution Environment nodes.

Toolbox icon Execution EnvironmentLearn more

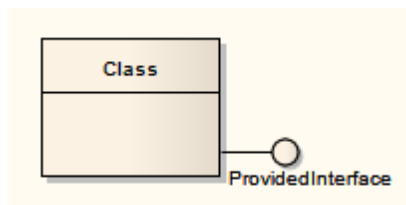
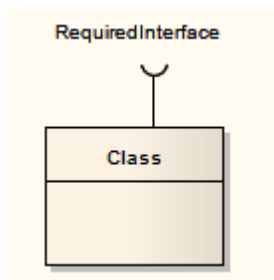
- [Deployment Diagram](#) ^[1020]
- [Node Element](#) ^[1196]
- [Component Element](#) ^[1187]
- [Artifact Element](#) ^[1179]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 210 states:

... an ExecutionEnvironment is ... usually part of a general Node, representing the physical hardware environment on which the ExecutionEnvironment resides. In that environment, the ExecutionEnvironment implements a standard set of services that Components require at execution time (at the modeling level these services are usually implicit). For each component Deployment, aspects of these services may be determined by properties in a DeploymentSpecification for a particular kind of ExecutionEnvironment.

6.4.2.12 Expose Interface



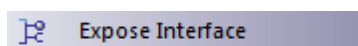
Description

The Expose Interface element is a graphical method of depicting the required or supplied interfaces of a Component, Class or Part, in a Component or Composite Structure diagram. It just identifies the fact that the element provides or requires an interface; to depict the fact that the provided interface is used, or the required interface provided, by another element using the Assembly connector.

The Expose Interface element must be attached to the Class or Component element, and it becomes a child element of that Class or Component; it cannot exist independently. You can attach more than one Expose Element to another element.

When you create the Expose Interface element, a dialog displays in which you enter a name for the element and specify whether it represents a required interface or a provided interface.

Toolbox icon

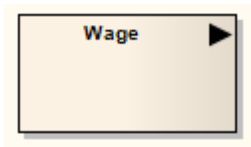


Learn more

- [Component Diagram](#) ^[1023]
- [Composite Structure Diagram](#) ^[1017]
- [Interface](#) ^[1194]
- [Component](#) ^[1187]

- [Class](#)^[1181]
- [Part](#)^[1200]
- [Assembly](#)^[1209]

6.4.2.13 Information Item

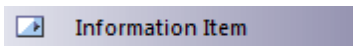


Description

An Information Item *element* represents an abstraction of data, which data can be conveyed between two objects. The *term* 'Information Item' is also more loosely applied to any classifier that represents a more specific identification of the type of data that can be conveyed between two objects

The conveyance and realization of Information Items (of either kind) between the two objects is represented by an Information Flow connector.

Toolbox icon



Learn more

- [Information Flow](#)^[1225]

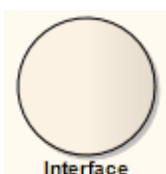
OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 608*) states:

An information item is an abstraction of all kinds of information that can be exchanged between objects. It is a kind of classifier intended for representing information at a very abstract way, one which cannot be instantiated.

One purpose of information items is to be able to define preliminary models, before having made detailed modeling decisions on types or structures. One other purpose of information items and information flows is to abstract complex models by a less precise but more general representation of the information exchanged between entities of a system.

6.4.2.14 Interface

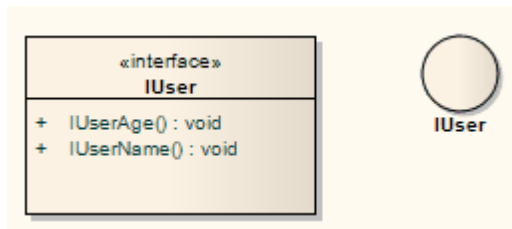


Description

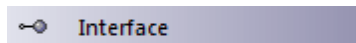
An *Interface* is a specification of behavior (or contract) that implementers agree to meet. By implementing an Interface, *Classes* are guaranteed to support a required behavior, which enables the system to treat non-related elements in the same way; that is, through the common interface. You also use Interfaces in a Composite Structure diagram.

Interfaces are drawn in a similar way to a Class, with operations specified, as shown below. They can also be drawn as a circle with no explicit operations detailed - right-click on the element and select the **Use Circle Notation** context menu option to switch between styles. Realize connectors to an Interface drawn as a circle are drawn as a solid line without target arrows.

An Interface cannot be instantiated (that is, you cannot create an object from an Interface). You must create a Class that 'implements' the Interface specification, and in the Class body place operations for each of the Interface operations. You can then instantiate the Class.



Toolbox icon



Learn more

- [Class Diagram](#) ^[1014]
- [Composite Structure Diagram](#) ^[1017]
- [Realization](#) ^[1254]

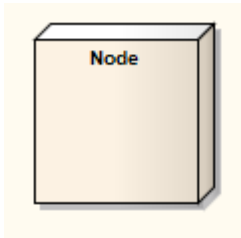
OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 88*) states:

An interface is a kind of classifier that represents a declaration of a set of coherent public features and obligations. An interface specifies a contract; any instance of a classifier that realizes the interface must fulfill that contract. The obligations that may be associated with an interface are in the form of various kinds of constraints (such as pre- and post-conditions) or protocol specifications, which may impose ordering restrictions on interactions through the interface.

Since interfaces are declarations, they are not instantiable. Instead, an interface specification is implemented by an instance of an instantiable classifier, which means that the instantiable classifier presents a public facade that conforms to the interface specification. Note that a given classifier may implement more than one interface and that an interface may be implemented by a number of different classifiers.

6.4.2.15 Node



Description

A Node is a physical piece of equipment on which the system is deployed, such as a workgroup server or workstation. A Node usually hosts components and other executable pieces of code, which again can be connected to particular processes or execution spaces. Typical Nodes are client workstations, application servers, mainframes, routers and terminal servers.

Nodes are used in Deployment diagrams to model the deployment of a system, and to illustrate the physical allocation of implemented artifacts. They are also used in web modeling, from dedicated web modeling pages in the Toolbox.

Toolbox icon



Learn more

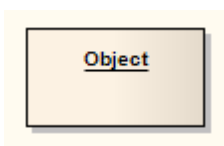
- [Deployment Diagram](#) 1020

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 213*) states:

In the metamodel, a Node is a subclass of Class. It is associated with a Deployment of an Artifact. It is also associated with a set of Elements that are deployed on it. This is a derived association in that these PackageableElements are involved in a Manifestation of an Artifact that is deployed on the Node. Nodes may have an internal structure defined in terms of parts and connectors associated with them for advanced modeling applications.

6.4.2.16 Object



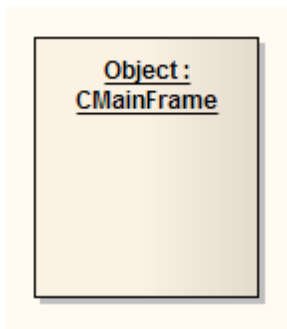
Description

An Object is a particular instance of a Class at run time. For example a car with the license plate AAA-001 is

an instance of the general class of cars with a license plate number attribute. Objects are often used in analysis to represent the numerous artifacts and items that exist in any business, such as pieces of paper, faxes and information. To model the varying behavior of Objects at run-time, use run-time states.

Early in analysis, Objects can be used to quickly capture all the things that are of relevance within the system domain, in an Object, Composite Structure or Communication diagram. As the model progresses these analysis Objects are refined into generic Classes from which instances can be derived to represent common business items. Once Classes are defined, Objects can be typed; that is they can have a classifier set that indicates their base type - see the *Object Classifiers* topic.

Enterprise Architect also supports a number of stereotyped Object elements to represent various entities in business modeling.



Toolbox icon



Learn more

- [Object Diagram](#) ^[1015]
- [Composite Structure Diagram](#) ^[1017]
- [Communication Diagram](#) ^[1084]
- [Class](#) ^[1181]
- [Object Classifiers](#) ^[898]
- [Business Interaction Objects](#) ^[1514]
- [Run-time State](#) ^[1197]
- [Define a Run-time Variable](#) ^[1198]
- [Remove a Defined Variable](#) ^[1198]
- [Object State](#) ^[1199]

6.4.2.16.1 Run-time State

At run-time, an Object instance can have specific values for its attributes, or exist in a particular state. To model the varying behavior of Objects at run-time, use instance values selected from the Select <Item> dialog and run-time states or run-states.

Typically there is interest in the run-time behavior of Objects that already have a classifier set. You can select from the classifier's attribute list and apply specific values for your Object instance. If the classifier has a child State Machine, its States propagate to a list where the run-time state for the Object can be defined.

Example

The following example defines run-time values for the listed variables, which are attributes of the instances' classifier *AccountItem*.



Learn more

- [Object](#)^[1196]
- [Object State](#)^[1199]
- [Define a Run-Time Variable](#)^[1198]
- [Remove a Defined Variable](#)^[1198]

6.4.2.16.1 Define a Run-time Variable

How to

To add run-time state instance variables to an Object

Step	Action
1	Right-click on the Object and, if Instance Variables are supported, select the Advanced Set Run State menu option (or press (Ctrl+Shift+R)). The Set Run State dialog displays.
2	In the Variable field, click on the drop-down arrow and select the variable, or type in the new variable name.
3	Set the Operator , the Value and optionally type in a Note .
4	Click on the OK button to save the variable.

6.4.2.16.1.2 Remove a Defined Variable

How to

To delete a run-time state variable for an Object

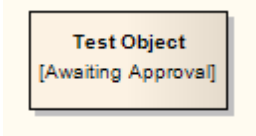
Step	Action
1	Right-click on the required Object and select the Set Run State context menu option. The Run State dialog displays.
2	In the Variable field, click on the drop-down arrow and select the variable to delete.
3	Clear the Value field.
4	Click on the OK button.

6.4.2.16.2 Object State

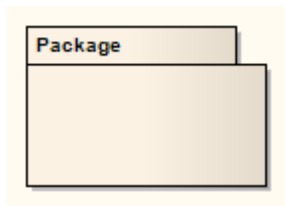
How to

To set the Object state for a Class instance

Step	Action
1	Right-click on the required Object and select the Advanced Set Object State context menu option. The Set Instance State dialog displays.
2	In the State field, either type the required State (such as Awaiting Approval) or select a State from the drop-down list. The drop-down list for the State field is populated with: <ul style="list-style-type: none"> • Any States owned by the object's classifier • Any States owned by any superclasses of the object's classifier • Any States owned by State Machines owned by the object's classifier • Any States owned by State Machines owned by any superclasses of the object's classifier
3	Click on the OK button to apply the State. The object now shows the run-time state in square brackets below the object name.

Step	Action
	

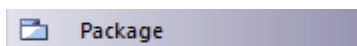
6.4.2.17 Package



Description

A *Package* is a namespace as well as an element that can be contained in other *Package*'s namespaces. A *Package* can own or merge with other *Packages*, and its elements can be imported into a *Package*'s namespace. In addition to using *Packages* in the Project Browser to organize your project contents, you can drag the *Packages* onto a diagram workspace (most diagram types, both standard and extended) for structural or relational depictions, including *Package* imports or merges.

Toolbox icon

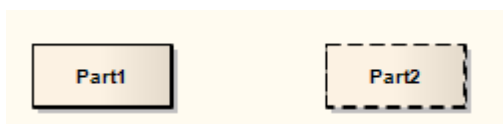


OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 109*) states:

A package is a namespace for its members, and may contain other packages. Only packageable elements can be owned members of a package. By virtue of being a namespace, a package can import either individual members of other packages, or all the members of other packages. In addition a package can be merged with other packages.

6.4.2.18 Part



Description

Parts are run-time instances of *Classes* or *Interfaces*. Multiplicity can be specified for a *Part*, using the notation:

(x{ . . . y })

where x specifies the initial or set number of instances when the composite structure is created, and y indicates the maximum number of instances at any time.

Parts are used to express composite structures, or modeling patterns that can be invoked by various objects to accomplish a specific purpose. When illustrating the composition of structures, Parts can be embedded as properties of other Parts. When embedded as properties, Parts can be bordered by a solid outline, indicating the surrounding Part owns the Part by composition. Alternatively, a dashed outline indicates that the property is referenced and used by the surrounding Part, but is not composed within it.

Toolbox icon



Learn more

- [Composite Structure Diagram](#) ^[1017]
- [Class](#) ^[1181]
- [Interface](#) ^[1194]
- [Properties](#) ^[1019]
- [Add Property Value](#) ^[1201]

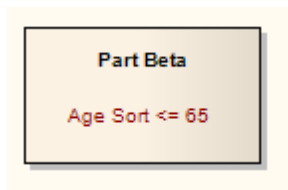
6.4.2.18.1 Add Property Value

How to

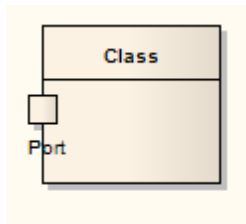
To add property value variables to a Part

Step	Action
1	Right-click on the Part and select the Advanced Set Property Values context menu option (or press (Ctrl+Shift+R)). The Set Property Values dialog displays.
2	In the Variable field, click on the drop-down arrow and select the variable, or type in the new variable name.
3	Set the Operator , the Value and optionally type in a Note .
4	Click on the OK button to save the variable.

A Part with a property value resembles the following figure:



6.4.2.19 Port



Description

Ports define the interaction between a classifier and its environment. *Interfaces* controlling this interaction can be depicted using the Interface element. Any connector to a Port must provide the required interface, if defined. Ports can appear on a contained Part, a Class, or the boundary of a Composite element.

A Port is a *typed* structural feature or property of its containing classifier. Ports are typically created in Class diagrams, Object diagrams and Composite Structure diagrams.

Toolbox icon



Learn more

- [Add a Port to an Element](#)^[1203]
- [Inherited and Redefined Ports](#)^[1203]
- [The Property Tab](#)^[1204]
- [Composite Structure Diagram](#)^[1017]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 182*) states:

A port is a property of a classifier that specifies a distinct interaction point between that classifier and its environment or between the (behavior of the) classifier and its internal parts. Ports are connected to properties of the classifier by connectors through which requests can be made to invoke the behavioral features of a classifier. A Port may specify the services a classifier provides (offers) to its environment as well as the services that a classifier expects (requires) of its environment.

6.4.2.19.1 Add a Port to an Element

How to

To add a new Port to an element, use *one* of the following steps:

Step	Action
1	Click on the Port symbol in the Composite Elements page of the Toolbox, and drag the symbol to (or click on) the target host element. This creates an untyped, simple Port on the boundary, near the cursor position.
2	On the context menu of a suitable Class, Part or Composite element, select the Embedded Elements Add Port option to add a new Port at the cursor position.
3	Drag a suitable classifier from the Project Browser onto a Class or Part. Enterprise Architect prompts you to add a typed Port or Part at the cursor position. The new Port is typed by the original dragged classifier.
4	Use the Embedded Elements window to add a new Port to the currently selected element.

Learn more

- [Composite Elements](#) ^[825]
- [Inherited and Redefined Ports](#) ^[1203]

6.4.2.19.2 Inherited and Redefined Ports

A Port is a *redefinable* and *re-useable* property of a composite classifier such as a Component. A Component can inherit Ports from its parent; if a Component's parent owns Ports, when you open the Embedded Elements dialog for the Component and select the **Show Owned/Inherited** checkbox the inherited Ports and their named owners are listed.

If you want to show an inherited Port on a Component, the Embedded Elements dialog provides two options:

- Expose an inherited Port - tick the checkbox next to the Port's name, to create a read-only copy of the Port; this is convenient for modeling Port interactions in child elements where the Ports are defined in the parent elements
- Redefine an inherited Port - select the row for the Port and click on the **Redefine** button, to create an editable copy of the Port; this is useful where a child element places additional restrictions or behavior on the Port

The above is also true for Components that inherit Ports from realized Interfaces, and for Component instances that inherit Ports from their classifying Component.

6.4.2.19.3 The Property Page

The element Properties dialog for Ports and Parts has a Property page in place of the Class element Details page.

This page defines the type, initial value, Qualifiers, multiplicity, and redefined and subsetted properties of the Port or Part.

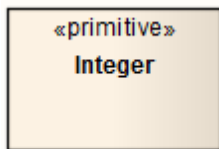
You set the Qualifiers by clicking on the **Qualifiers** button, to display the Qualifiers dialog.

You add **Redefined** and **Subsetted Properties** by clicking on the appropriate **Add** button, to display the Select Property dialog.

Learn more

- [Qualifiers Dialog](#)^[1212]
- [Select Property Dialog](#)^[882]
- [Inherited and Redefined Ports](#)^[1203]

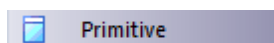
6.4.2.20 Primitive



Description

A Primitive element identifies a predefined data type, without any relevant substructure (that is, it has no parts in the context of UML). It could be regarded as a conceptual Data Type.

Toolbox icon



Learn more

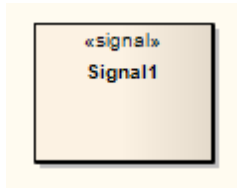
- [Data Type Element](#)^[1188]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 124*) states:

A primitive data type may have an algebra and operations defined outside of UML, for example, mathematically ... The run-time instances of a primitive type are data values. The values are in many-to-one correspondence to mathematical elements defined outside of UML (for example, the various integers). Instances of primitive types do not have identity. If two instances have the same representation, then they are indistinguishable.

6.4.2.21 Signal



Description

A Signal is a specification of Send request instances communicated between objects, typically in a Class or Package diagram. The receiving object handles the Received request instances as specified by its receptions. The data carried by a Send request is represented as attributes of the Signal. A Signal is defined independently of the classifiers handling the signal occurrence.

To define a reception, create an operation in the receiving object and assign the stereotype <<signal>> to it. The reception has the same name as the signal that the object can receive.

Toolbox icon



Learn more

- [Class Diagram](#)^[1014]
- [Package Diagram](#)^[1017]
- [Send Element](#)^[1157]
- [Receive Element](#)^[1149]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 450*) states:

A signal triggers a reaction in the receiver in an asynchronous way and without a reply. The sender of a signal will not block waiting for a reply but continue execution immediately. By declaring a reception associated to a given signal, a classifier specifies that its instances will be able to receive that signal, or a subtype thereof, and will respond to it with the designated behavior.

And (*UML Superstructure Specification, v2.1.1, p. 447 - 448*):

A reception is a declaration stating that a classifier is prepared to react to the receipt of a signal. A reception designates a signal and specifies the expected behavioral response. The details of handling a signal are specified by the behavior associated with the reception or the classifier itself. ...Receptions are shown using the same notation as for operations with the keyword <signal>



6.5 UML Connectors

A *connector* is a logical or functional relationship between model elements. There are several different connector types, each having a particular purpose and syntax. Enterprise Architect supports all of the UML connectors as well as various custom connectors. Together with the UML Elements, these form the basis of UML models.

For more information on using these connectors, consult the appropriate topic by clicking on the required connector icon in the table below.

Behavioral Diagram Connectors	Structural Diagram Connectors	Inbuilt and Extended Connectors
<p>Activity Diagrams</p> <ul style="list-style-type: none"> Control Flow Object Flow Interrupt Flow <p>Use Case Diagrams</p> <ul style="list-style-type: none"> Use Associate Generalize Include Extend Realize Invokes Precedes <p>State Diagrams</p> <ul style="list-style-type: none"> Transition Object Flow <p>Timing Diagrams</p>	<p>Composite Structure Diagrams</p> <ul style="list-style-type: none"> Connector Assembly Delegate Role Binding Represents Occurrence <p>Package and Class Diagrams</p> <ul style="list-style-type: none"> Associate Generalize Compose Aggregate Association Class Assembly Realize Nesting Package Merge Package Import 	<p>Analysis Diagrams</p> <ul style="list-style-type: none"> Information Flow Object Flow Associate Realize Representation <p>Common Connectors</p> <ul style="list-style-type: none"> Dependency Realize Trace Information Flow Note Link <p>Profile</p> <ul style="list-style-type: none"> Extension Generalize Application Tagged Value Redefinition

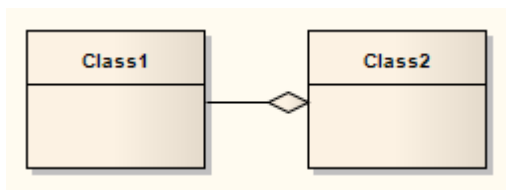
Behavioral Diagram Connectors	Structural Diagram Connectors	Inbuilt and Extended Connectors
<p>⇒ Message</p>	<p>Component Diagrams</p> <p>⊖ Assembly</p> <p>↗ Delegate</p> <p>↗ Associate</p> <p>↗ Realize</p> <p>↗ Generalize</p>	<p>Metamodel</p> <p>↗ Generalize</p> <p>↗ Associate</p> <p>↗ Compose</p> <p>↗ Aggregate</p>
<p>Sequence Diagrams</p> <p>⇒ Message</p> <p>↻ Self-Message</p> <p>⏏ Recursion</p> <p>⏏ Call</p>	<p>Deployment Diagrams</p> <p>↗ Associate</p> <p>↗ Communication Path</p> <p>↗ Association Class</p> <p>↗ Generalize</p> <p>↗ Realize</p> <p>↗ Deployment</p> <p>↗ Manifest</p> <p>↗ Nesting</p>	<p>Custom</p> <p>↗ Associate</p> <p>↗ Aggregate</p> <p>↗ Generalize</p> <p>↗ Realize</p> <p>↗ Nesting</p>
<p>Communication Diagrams</p> <p>↗ Associate</p> <p>↗ Realize</p> <p>↗ Nesting</p>	<p>User Interface</p> <p>↗ Associate</p> <p>↗ Aggregate</p> <p>↗ Generalize</p> <p>↗ Realize</p>	<p>Requirements</p> <p>↗ Aggregate</p> <p>↗ Inheritance</p> <p>↗ Associate</p> <p>↗ Implements</p>
<p>Interaction Overview Diagrams</p> <p>↗ Control Flow</p> <p>↗ Object Flow</p> <p>↗ Interrupt Flow</p>	<p>XML Schema</p> <p>↗ Generalize</p> <p>↗ Associate</p>	<p>WSDL</p> <p>No special connectors</p>
<p>Maintenance</p> <p>↗ Aggregate</p>	<p>Object</p> <p>↗ Information Flow</p> <p>↗ Associate</p>	<p>Documentation</p> <p>No special connectors</p>
<p>Data Modeling</p>		

Behavioral Diagram Connectors	Structural Diagram Connectors	Inbuilt and Extended Connectors
 Associate	 Dependency	

Notes

- Invokes and Precedes relationships are defined by the Open Modeling Language (OML); they are stereotyped Dependency relationships - Invokes indicates that Use Case A, at some point, causes Use Case B to happen, whilst Precedes indicates that Use Case C must complete before Use Case D can begin
- An Extension relationship shows that a stereotype extends one or more metaclasses; all stereotypes must extend either one or more Metaclasses, or another stereotype that extends a stereotype (that itself extends a stereotype, and so on)
- A Tagged Value relationship defines a reference-type (that is, RefGUID) Tagged Value owned by the source stereotype; the Tagged Value is named for the target role of this association, and is limited to referencing elements with the stereotype by the association target element
- The Application and Redefinition relationships are **deprecated**

6.5.1 Aggregation

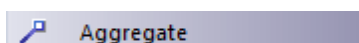


Description

An Aggregation connector is a type of association that shows that an element contains or is composed of other elements. It is used in Class models, Package models and Object models to show how more complex elements (aggregates) are built from a collection of simpler elements (component parts; for example, a car from wheels, tires, motor and so on).

A stronger form of aggregation, known as Composite Aggregation, is used to indicate ownership of the whole over its parts. The part can belong to only one Composite Aggregation at a time. If the composite is deleted, all of its parts are deleted with it.

Toolbox icon



Learn more

- [Class Diagram](#)^[1014]
- [Package Diagram](#)^[1017]

- [Object Diagram](#) ^[1015]
- [Change Aggregation Connector Form](#) ^[1209]

6.5.1.1 Change Aggregation Connector Form

In Enterprise Architect, the default Aggregation relationship is the weak form of the relationship, represented by a hollow diamond.

How to

To change the form of an Aggregation connector from weak to strong

Step	Action
1	Right-click on an Aggregation connector to display the context menu.
2	Select Set Aggregation to Composite . The diamond is shown as filled.

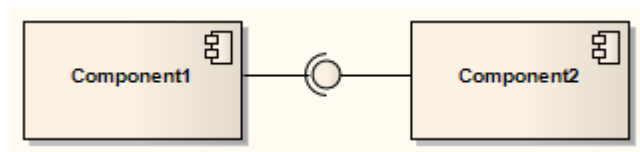
Notes

- If the connector is already a Strong (Composition) connector, the context menu option changes to **Set Aggregation to Shared**

Learn more

- [Aggregation](#) ^[1208]

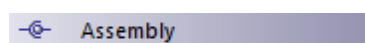
6.5.2 Assembly



Description

An Assembly connector bridges a component's required interface (*Component1*) with the provided interface of another component (*Component2*), typically in a Component diagram.

Toolbox icon



Learn more

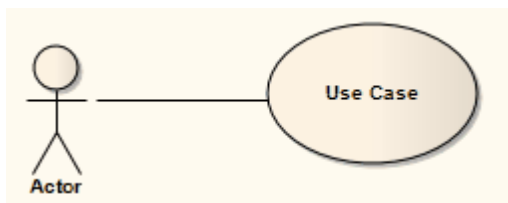
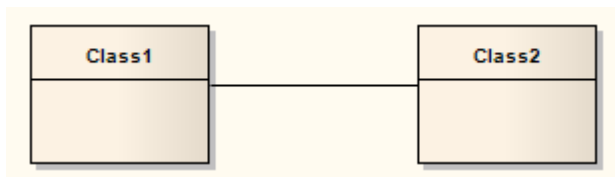
- [Component Diagram](#) ^[1023]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 156*) states:

An assembly connector is a connector between two components that defines that one component provides the services that another component requires. An assembly connector is a connector that is defined from a required interface or port to a provided interface or port.

6.5.3 Association



Description

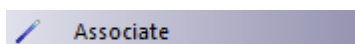
An Association implies two model elements have a relationship, usually implemented as an instance variable in one Class. This connector can include named roles at each end, multiplicity, direction and constraints. Association is the general relationship type between elements. To connect more than two elements in an association, you can use the *N-Ary Association* element.

When code is generated for Class diagrams, Associations become instance variables in the target Class. The relationship is also used in Package, Object, Communication, Data Modeling and Deployment diagrams.

An Associate connector can also be integrated with a Class element to form an Association Class, to allow the Associate connector to have operations and attributes that define certain types of UML relationship.

You can also define template binding parameters for an Associate connector between a binding Class and a parameterized Class.

Toolbox icon



Learn more

- [N-Ary Association](#) ^[1673]
- [Association Class](#) ^[1215]

- [Class Diagrams](#)^[1014]
- [Template Binding](#)^[1257]
- [Qualifiers](#)^[1217]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 41*) states:

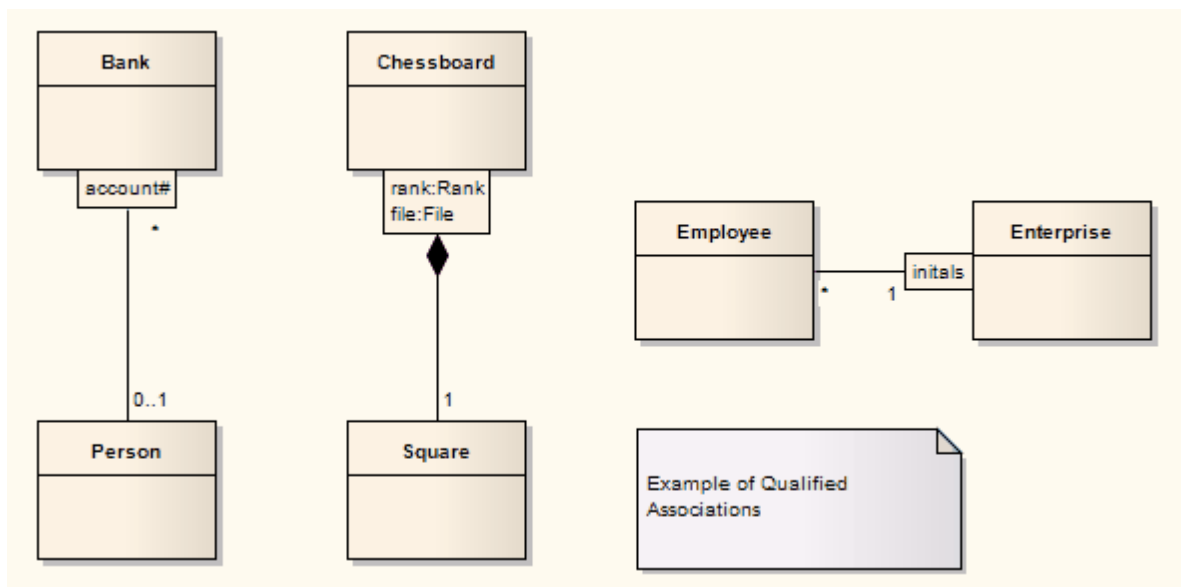
An association specifies a semantic relationship that can occur between typed instances. It has at least two ends represented by properties, each of which is connected to the type of the end. More than one end of the association may have the same type.

An end property of an association that is owned by an end class or that is a navigable owned end of the association indicates that the association is navigable from the opposite ends; otherwise, the association is not navigable from the opposite ends.

6.5.3.1 Qualifiers

Qualifiers are ordered sets of properties of an Association end point, a Part, a Port, or an Attribute, that limit the nature of the relationship between two classifiers or objects. You define a qualifier on the Qualifiers dialog, which you display by clicking on the (...) button at the end of the **Qualifiers** field on the Association, Part, Port or Attribute Properties dialog.

Examples



Notes

- When typing multiple Qualifiers into the **Qualifier(s)** field on a Properties dialog, separate them with a semi-colon; each Qualifier then displays on a separate line - for example, in the diagram the Qualifier `'rank:Rank;file:File'` has been rendered in two lines, with a line break at the ; character
- You can enable or disable Qualifier rectangles in the Diagram page of the Options dialog (select the **Tools | Options | Diagram** menu option) - if disabled, the old style text Qualifiers are used; it is not

recommended that you disable Qualifiers as they are an integral part of the UML

- You can enable or disable a mild shading on the Qualifier rectangles in the Links page of the Options dialog

Learn more

- [Qualifiers Dialog](#)^[1212]
- [Source Role](#)^[964]
- [Port](#)^[1202]
- [Part](#)^[1200]
- [Attribute](#)^[885]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 129*) states:

A qualifier declares a partition of the set of associated instances with respect to an instance at the qualified end (the qualified instance is at the end to which the qualifier is attached). A qualifier instance comprises one value for each qualifier attribute. Given a qualified object and a qualifier instance, the number of objects at the other end of the association is constrained by the declared multiplicity. In the common case in which the multiplicity is 0..1, the qualifier value is unique with respect to the qualified object, and designates at most one associated object. In the general case of multiplicity 0.., the set of associated instances is partitioned into subsets, each selected by a given qualifier instance. In the case of multiplicity 1 or 0..1, the qualifier has both semantic and implementation consequences. In the case of multiplicity 0..*, it has no real semantic consequences but suggests an implementation that facilitates easy access of sets of associated instances linked by a given qualifier value.*


6.5.3.1.1 Qualifiers Dialog

The Qualifiers dialog is used to define the Qualifiers of an Association connector end, Port, Part or Attribute.

General Tab:

Review, edit or complete the fields as indicated in the following table:

Field	Action	See also
Name	Display the name of the Qualifier. For a new Qualifier, type the name (with no spaces).	
Alias	Display an optional alias for the Qualifier. If necessary, type in a new alias.	
Type	Display the Qualifier type. The type can be defined by the code language (data type) or by a classifier element. When you click on the drop-down arrow, the set of values in the list provides the appropriate data types.	Select <Item> Dialog ^[880] Data Types ^[987]

Field	Action	See also
	<p>To select or define possible classifiers, either click on the Select Type option in the list, or click on the (...) (Select) button to display the Select <Item> dialog.</p> <p>To add new code language data types that can be displayed in this list, see the <i>Data Types</i> topic.</p>	
Scope	<p>Define the Qualifier as Public, Protected, Private or Package.</p> <p>If necessary, click on the drop-down arrow and select a different scope.</p>	
Stereotype	<p>Define the optional stereotype of the Qualifier.</p> <p>If necessary, either type a different stereotype name or click on the drop-down arrow and select a stereotype.</p>	
Derived	<p>Indicate that the Qualifier is a calculated value.</p> <p>If you select this checkbox, the Qualifier name on the element has the derived symbol (/) as a prefix.</p>	
Static	<p>Indicate that the Qualifier is a static member.</p>	
Const	<p>Indicate that the Qualifier is a constant.</p>	
Initial	<p>Display an optional initial value.</p> <p>If necessary, type in a new initial value.</p>	
Notes	<p>Enter any free text notes associated with the Qualifier.</p> <p>You can format the notes text using the Notes toolbar at the top of the field.</p>	Notes  toolbar

To change the position of a Qualifier in the list in the Qualifiers panel, click on the **Scroll Up** or **Scroll Down** (hand) buttons.

Detail Tab:

Use the Detail tab to model additional properties of a selected Qualifier, such as its multiplicity, redefined properties and subsetted properties.

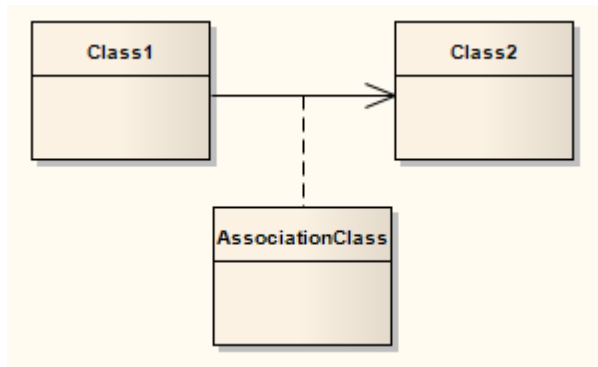
Select a Qualifier on the General tab, then review, edit or complete the Detail tab fields as indicated in the following table.

Field	Action	See also
Multiplicity		
Lower bound	Define a lower limit to the number of elements allowed in the collection.	
Upper bound	Define an upper limit to the number of elements allowed in the collection.	
Allow Duplicates	Indicate that duplicates are allowed. Maps to the UML property <i>isUnique</i> , value <i>FALSE</i> .	
Multiplicity is Ordered	Indicate that the collection is ordered.	
Redefined Property	Review the redefined properties for the Qualifier. Add redefined properties by clicking on the Add button to display the Select Property dialog.	Select Property Dialog ^[882]
Subsetted Property	Review the subsetted properties for the qualifier. Add subsetted properties by clicking on the Add button to display the Select Property dialog.	Select Property Dialog ^[882]

Learn more

- [Qualifiers](#) ^[1211]
- [Source Role](#) ^[964]
- [Port](#) ^[1202]
- [Part](#) ^[1200]
- [Attribute](#) ^[885]

6.5.4 Association Class



Description

An *Association Class* is a UML construct that enables an Association to have *attributes* and *operations* (*features*). This results in a hybrid relation with the characteristics of an Association and a Class.

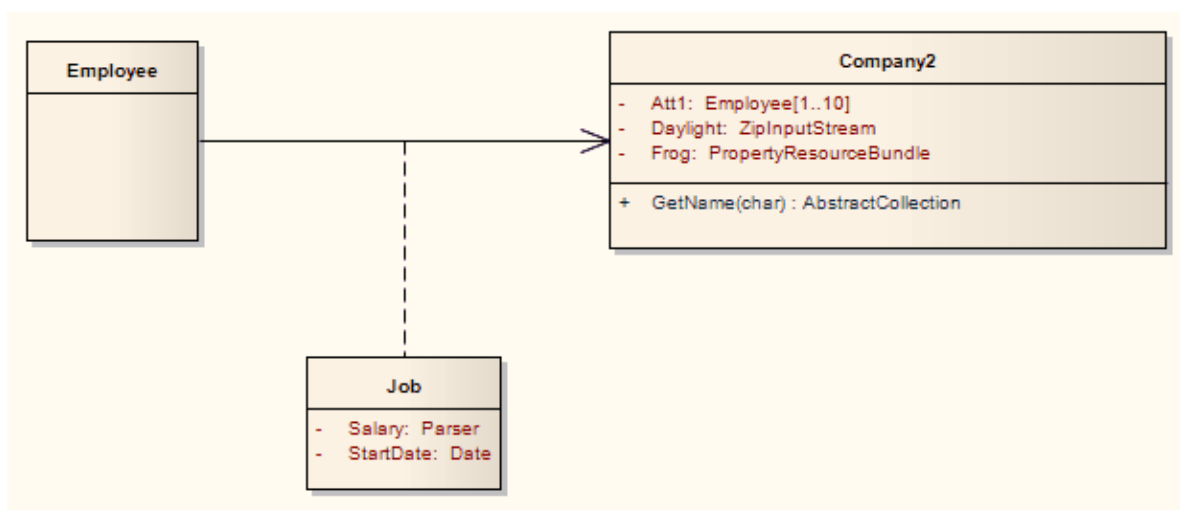
When you add an Association Class connection, Enterprise Architect also creates a Class that is automatically connected to the Association. When you hide or delete the Association, the Class is also hidden or deleted.

To add an Association Class to a Class or Deployment diagram, click on the *Association Class* icon in the Toolbox. Click and hold on the source object in the diagram while you drag the line to the target element, then release the mouse button. Enterprise Architect draws the connector and adds the Class, then prompts you to add the Class name. Note that the names of the Class and the connector are the same. You can also connect a new Class to an existing Association.

You can highlight the Class part of an Association Class in the Project Browser, by selecting the **Find Association Class** context menu option on the Association connector.

Example

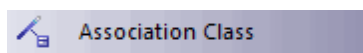
The following diagram illustrates an Association Class between model elements. Note the dotted line from the Class to the Association. You cannot move or delete this line.



Notes

- If you are applying a stereotype with a Shape Script to an Association Class, be aware that the Shape Script is applied to both the Class part and the Association part; therefore, you might have to include logic in the *shape main* that tests the type of the element so that you can give separate drawing instructions for Class and for Association
- Such logic is not necessary in the:
 - Shape source or shape target, which are ignored by Classes, or the
 - Decoration shapes, which are ignored by Association connectors
- If you dissociate the Class from the Association connector, both parts keep their Shape Scripts until the stereotypes are removed.

Toolbox icon



Learn more

- [Class Diagram](#)^[1014]
- [Connect New Class to Association](#)^[1216]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 49*) states:

A model element that has both association and class properties. An AssociationClass can be seen as an association that also has class properties, or as a class that also has association properties. It not only connects a set of classifiers but also defines a set of features that belong to the relationship itself and not to any of the classifiers.

6.5.4.1 Connect New Class to Association

How to

To connect a new Class to an existing Association

Step	Action
1	Create a Class in the diagram containing the Association to connect.
2	Right-click on the new Class and select the Advanced Association Class menu option. The Create Association Class dialog displays.

Step	Action
3	Select the connector to connect to.
4	Click on the OK button.

Learn more

- [Association Class](#) [1215]

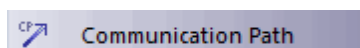
6.5.5 Communication Path



Description

A *Communication Path* defines the path through which two *DeploymentTargets* are able to exchange signals and messages. Communication Path is a specialization of *Association*. A *DeploymentTarget* is the target for a deployed *Artifact* and can be a *Node*, *Property* or *InstanceSpecification* in a Deployment diagram.

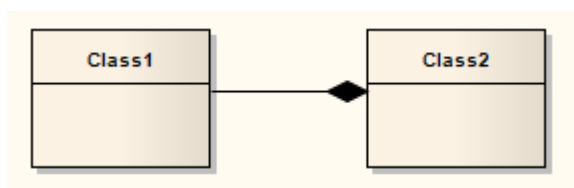
Toolbox icon



Learn more

- [Deployment Diagram](#) [1020]
- [Association](#) [1210]

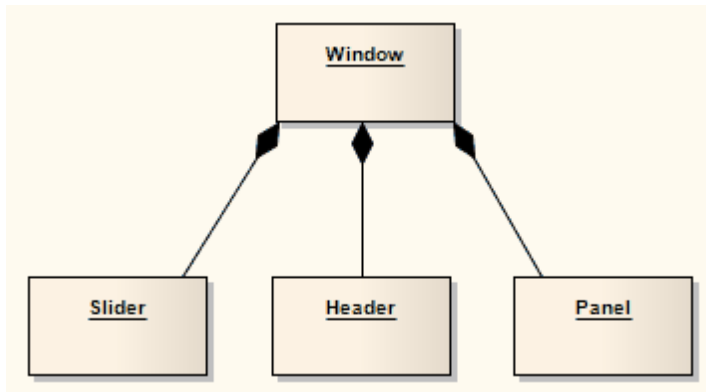
6.5.6 Composition



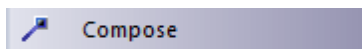
Direction:

A Composition is used to depict an element that is made up of smaller components, typically in a Class or Package diagram. A component - or part instance - can be included in a maximum of one composition at a time. If a composition is deleted, usually all of its parts are deleted with it; however, a part can be individually removed from a composition without having to delete the entire composition. Compositions are transitive, asymmetric relationships and can be recursive.

Example



Toolbox icon



Learn more

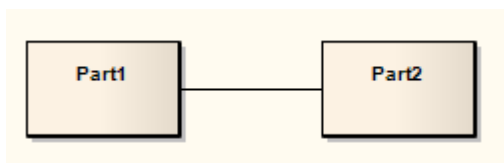
- [Aggregation](#) ^[1208]
- [Class Diagram](#) ^[1014]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 43*) states:

Composite aggregation is a strong form of aggregation that requires a part instance be included in at most one composite at a time. If a composite is deleted, all of its parts are normally deleted with it.

6.5.7 Connector

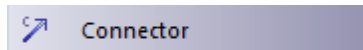


Description

Connectors illustrate communication links between Parts to fulfill the structure's purpose, typically in a

Composite Structure diagram. Each Connector end is distinct, controlling the communication pertaining to its connecting element. These elements can define constraints specifying this behavior. Connectors can have multiplicity.

Toolbox icon



Learn more

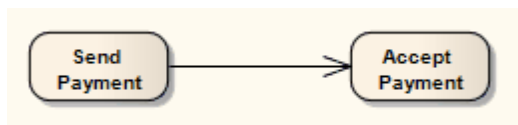
- [Composite Structure Diagram](#) 1017

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 177*) states:

Specifies a link that enables communication between two or more instances. This link may be an instance of an association, or it may represent the possibility of the instances being able to communicate because their identities are known by virtue of being passed in as parameters, held in variables or slots, or because the communicating instances are the same instance. The link may be realized by something as simple as a pointer or by something as complex as a network connection. In contrast to associations, which specify links between any instance of the associated classifiers, connectors specify links between instances playing the connected parts only.

6.5.8 Control Flow



Description


The *Control Flow* is a connector connecting two nodes in an Activity diagram. Control Flow connectors bridge the flow between Activity nodes, by directing the flow to the target node once the source node's activity is completed.

Control Flows and Object Flows can define a *Guard* and a *Weight* condition.

A *Guard* defines a condition that must be true before control passes along that activity edge. A practical example of this is where two or more activity edges (Control Flows) exit from a Decision element. Each flow should have a *Guard* condition that is exclusive of the other and defines which edge is taken under what conditions. The Control Flow Properties dialog enables you to set up *Guard* conditions on Control Flows and on Object Flows.

A *Weight* defines the number of tokens that can flow along a Control or Object Flow connection when that edge is traversed. *Weight* can also be defined on the Control Flow and Object Flow Properties dialogs.

Toolbox icon

 Control Flow**Learn more**

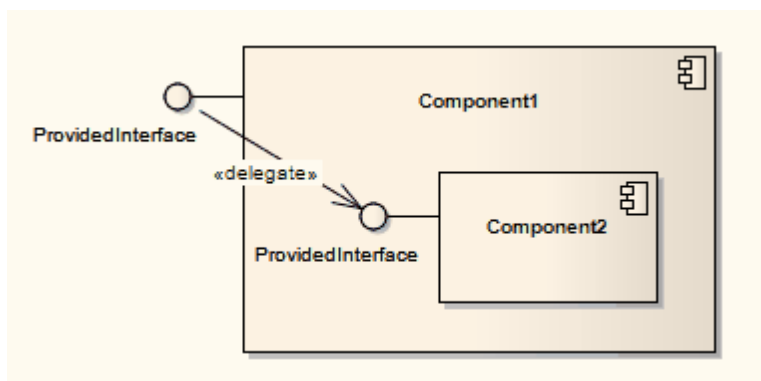
- [Activity diagram](#) ^[1028]
- [Object Flow](#) ^[1256]
- [Decision Node](#) ^[1117]

OMG UML specification:


The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 356*) states:

A control flow is an edge that starts an activity node after the previous one is finished.

6.5.9 Delegate

**Description**

A Delegate connector defines the internal assembly of a component's external Ports and Interfaces, on a Component diagram. Using a Delegate connector wires the internal workings of the system to the outside world, by a delegation of the external interfaces' connections.

Toolbox icon Delegate**Learn more**

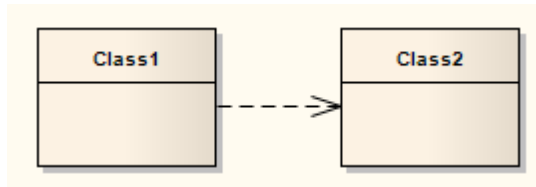
- [Component Diagram](#) ^[1023]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 156*) states:

A delegation connector is a connector that links the external contract of a component (as specified by its ports) to the internal realization of that behavior by the component's parts. It represents the forwarding of signals (operation requests and events): a signal that arrives at a port that has a delegation connector to a part or to another port will be passed on to that target for handling.

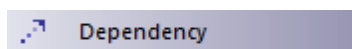
6.5.10 Dependency



Description

Dependency relationships are used to model a wide range of dependent relationships between model elements in Use Case, Activity and Structural diagrams, and even between models themselves. You can create the Dependency from the Common page of the Toolbox. The Dependencies package as defined in UML 2.3 has many derivatives, such as Realize, Deployment and Use. Once you create a Dependency you can further refine its meaning by applying a specialized stereotype.

Toolbox icon



Learn more

- [Realization](#) ^[1254]
- [Deployment](#) ^[1222]
- [Use](#) ^[1263]
- [Apply a Stereotype](#) ^[1221]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 64*) states:

A dependency is a relationship that signifies that a single or a set of model elements requires other model elements for their specification or implementation. This means that the complete semantics of the depending elements is either semantically or structurally dependent on the definition of the supplier element(s).

6.5.10.1 Apply a Stereotype

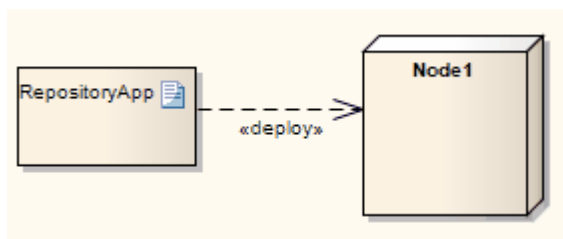
How to

To apply a stereotype to a Dependency relationship

Step	Action
1	Select the Dependency relationship to change.
2	Right-click on the connector and, from the context menu, select the Dependency Properties option. The Dependency Properties dialog displays.
3	In the Stereotype field, either type in the required stereotype name or click on the drop-down arrow and select the stereotype from the list.
4	Click on the OK button.

Alternatively, you can right-click on the Dependency relationship and select the **Advanced | Dependency Stereotypes** context menu option, then select from a shorter list of standard stereotypes.

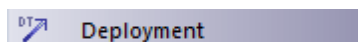
6.5.11 Deployment



Description

A Deployment is a type of Dependency relationship that indicates the deployment of an artifact onto a node or executable target, typically in a Deployment diagram. A Deployment can be made at type and instance levels. At the type level, a Deployment would be made for every instance of the node. Deployment can also be specified for an instance of a node, so that a node's instances can have varied deployed artifacts. With composite structures modeled with nodes defined as Parts, Parts can also serve as targets of a Deployment relationship.

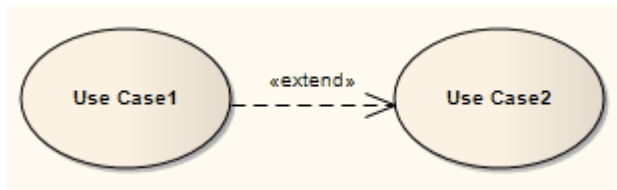
Toolbox icon



Learn more

- [Deployment Diagram](#) 

6.5.12 Extend



Description

An *Extend* connection is used to indicate that an element extends the behavior of another. Extensions are used in Use Case models to indicate that one Use Case (optionally) extends the behavior of another. An extending Use Case often expresses alternative flows.

Toolbox icon



Learn more

- [Use Case Diagram](#) f1030

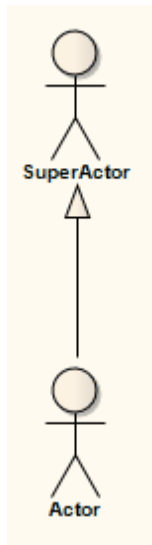
OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 587) states:

This relationship specifies that the behavior of a Use Case may be extended by the behavior of another (usually supplementary) Use Case. The extension takes place at one or more specific extension points defined in the extended Use Case. Note, however, that the extended Use Case is defined independently of the extending Use Case and is meaningful independently of the extending Use Case. On the other hand, the extending Use Case typically defines behavior that may not necessarily be meaningful by itself. Instead, the extending Use Case defines a set of modular behavior increments that augment an execution of the extended Use Case under specific conditions.

Note that the same extending Use Case can extend more than one Use Case. Furthermore, an extending Use Case may itself be extended.

6.5.13 Generalization

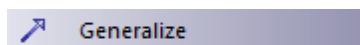


Description

A Generalization is used to indicate inheritance. Drawn from the specific classifier to a general classifier, the generalization's implication is that the source inherits the target's characteristics. It is used typically in Class, Component, Object, Package, Use Case and Requirements diagrams.

You can also define template binding parameters for a Generalize connector between a binding Class and a parameterized Class.

Toolbox icon

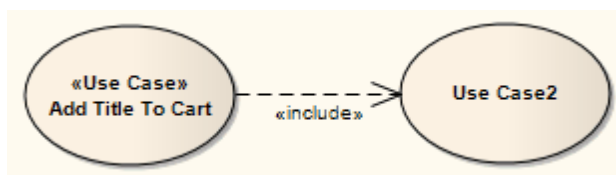


OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 73*) states:

A generalization is a taxonomic relationship between a more general classifier and a more specific classifier. Each instance of the specific classifier is also an indirect instance of the general classifier. Thus, the specific classifier inherits the features of the more general classifier.

6.5.14 Include



Description

An Include connection indicates that the source element includes the functionality of the target element. Include connections are used in Use Case models to reflect that one Use Case includes the behavior of another. Use an Include relationship to avoid having the same subset of behavior in many Use Cases; this is similar to delegation used in Class models.

Toolbox icon



Learn more

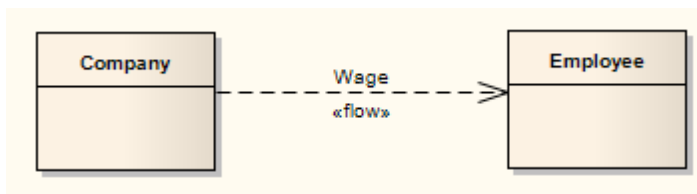
- [Use Case Diagram](#) [r030]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 591*) states:

Include is a DirectedRelationship between two Use Cases, implying that the behavior of the included Use Case is inserted into the behavior of the including Use Case. It is also a kind of NamedElement so that it can have a name in the context of its owning Use Case. The including Use Case may only depend on the result (value) of the included Use Case. This value is obtained as a result of the execution of the included Use Case.

6.5.15 Information Flow

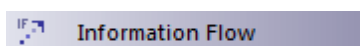


Description

An Information Flow represents the flow of Information Items (either Information Item elements or classifiers) between two elements in any diagram. The connector is available from the Common page of the Toolbox, from every Quick Link menu, and automatically whilst directly defining Information Item realization. When you create the connector, Enterprise Architect automatically prompts you to identify which information items are conveyed.

You can have more than one Information Flow connector between the same two elements, identifying which items flow between the elements under differing conditions. The connectors can flow in the same direction, or opposite directions.

Toolbox icon



Learn more

- [Information Item](#) ^[1194]
- [Using Information Flows](#) ^[1226]
- [Convey Information on a Flow](#) ^[1228]
- [Realize an Information Flow](#) ^[1229]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 606*) states:

An InformationFlow specifies that one or more information items circulates from its sources to its targets.

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 607*) also states:

An information flow is an abstraction of the communication of an information item from its sources to its targets. It is used to abstract the communication of information between entities of a system. Sources or targets of an information flow designate sets of objects that can send or receive the conveyed information item.

6.5.15.1 Using Information Flows

When you drag an Information Flow connector between two elements on a diagram, Enterprise Architect automatically prompts you to identify the Information items conveyed. You can also immediately go on to realize the Information Flows. However, you can also defer the identification of Information Items at either stage.

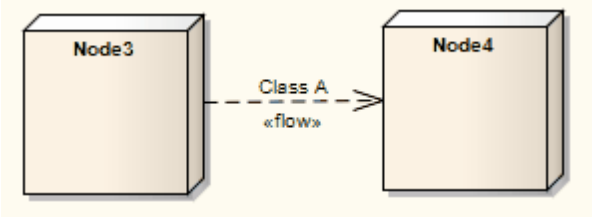
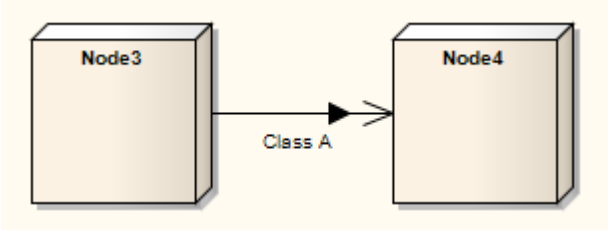
The procedure below describes the full process and identifies where to break off if required.

You can also create an Information Flow automatically whilst directly defining Information Flow realization.

How to

To create and realize Information Flows

Step	Action	See also
1	Open a diagram and add two elements (for example, Nodes on a Deployment diagram).	
2	Click on the <i>Information Flow</i> connector in the Common page of the Toolbox and drag the cursor between the two elements. The Information Items Conveyed dialog displays. (If you prefer, click on the Cancel button to create the connector without identifying any information items, to return to the process at a later time.)	Convey Information on a Flow ^[1228]

Step	Action	See also
3	<p>Add the classifier or Information Item element(s) to the Information Flow.</p> <p>The diagram now resembles the following:</p>  <p>(If you prefer, click on the OK button to leave the information flow without realizing the flow, to return to the process at a later time.)</p>	Convey Information on a Flow ^[1228]
4	<p>Add another connector between the same two elements (for example, a <i>Communication Path</i> connector).</p>	
5	<p>Right-click the connector and select the Advanced Information Flows Realized context menu option.</p> <p>The Information Flows Realized dialog displays.</p>	
6	<p>Tick the checkbox against each required information item in the realized flow and click on the OK button.</p> <p>The combined connector now resembles the following:</p> 	Realize an Information Flow ^[1229]

Notes

- Once the Information Flow is realized, you cannot access the Information Items Conveyed dialog directly; you add or delete information items on the connector using the Information Items Realized dialog
- If you have more than one Information Flow connector between the elements, they form part of the same combined connector; you can again work on them separately through the Information Items Realized dialog

- If you have information flows in a diagram that you use as the source for a Pattern, the Information Items Conveyed and Information Flows Realized data is **not** copied into the Pattern
- You can locate, in the Project Browser, the classifier or information item element(s) conveyed on the Information Flow connector, using the **Find Items Conveyed** context menu option on the connector

6.5.15.2 Convey Information on a Flow

When you create an Information Flow connector between two elements, Enterprise Architect automatically prompts you to specify which Information Items or classifiers are conveyed on this flow.

You can ignore the prompt and add the information items later, or - if you have not realized the information flow with its *existing* information items - you can change and/or add to the information items conveyed.

Access [Information Flow context menu | Advanced | Information Items Conveyed](#)

How to

To specify the Information Items conveyed on an Information Flow

Step	Action	See also
1	On the Information Items Conveyed dialog, click on the Add button. The Select Classifier dialog displays.	
2	Browse or search for the required Information Item or classifier element or elements and select them as required. If you do not want to retain a selected item on the Select Classifier dialog, press (Ctrl) and click on the item.	Select <Item> Dialog ^[880]
3	Click on the OK button to return to the Information Items Conveyed dialog. Each information item you have selected is listed on a separate line in the dialog.	
4	If you do not want to retain a selected item on this dialog, click on it and click on the Remove button. Click on the OK button to close the dialog and to show the selected information item element names on the Information Flow connector label. If appropriate, you can now realize the information items conveyed in this Information Flow.	Realize an Information Flow ^[1229]

Learn more

- [Information Flow](#) ^[1225]
- [Information Item](#) ^[1194]

- [Using Information Flows](#) 

6.5.15.3 Realize an Information Flow

Access **Connector context menu | Advanced | Information Flows Realized**

The Information Flows Realized dialog displays all existing flows that can be realized on the selected connector.

Use to

- Realize one or more existing flows on an Information Flow connector
- Edit an existing flow on an Information Flow connector
- Create and realize information flows on a non Information Flow connector

How to

Operation	Action
Realize Information Flows on the selected connector	Select the checkbox for each required flow and click on the OK button.
Cancel realization of a flow	Deselect the checkbox against the appropriate flow, and click on the OK button.
Change the classifier or Information Item elements conveyed on an Information Flow	<p>Click on the flow text, then click on the [...] button.</p> <p>The Select Classifier dialog displays:</p> <ul style="list-style-type: none"> • Click on a single item to select it • [Ctrl]+click on each of several items to select them all, or • [Ctrl]+click on a selected item to deselect it <p>Click on the OK button to return to the Information Flows Realized dialog and, if required, realize the changed flow as above.</p>
Create a realized information flow connector on another connector	<ol style="list-style-type: none"> 1. Right-click on the connector and select the Information Flows Realized context menu option. 2. Click on the <i>Click to create new information flow...</i> text. The Select Classifier dialog displays. 3. Select the required classifier or Information Item elements, and click on the OK button to return to the Information Flows Realized dialog; the selected elements are listed first on the dialog, with the activation checkbox ticked. 4. Click on the OK button to return to the diagram; the connector now displays as a realized Information Flow, with the selected

Operation	Action
	classifier or Information Item elements named in the connector label.

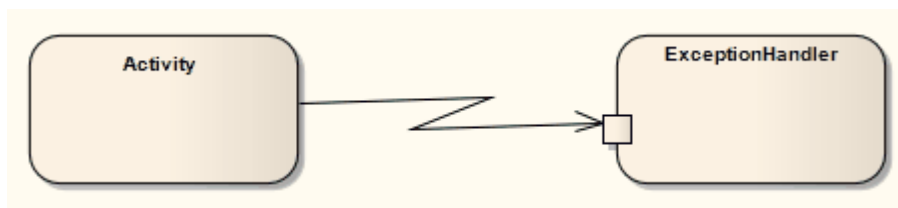
Notes

- If there are several Information flows and you do not realize all of them, those that are not realized are represented by a separate Information Items Conveyed iteration of the Information Flow connector; you can only realize those flows on the *original* connector, at which point the flow is represented on that original connector
- If you realize all of the flows, they are combined on the one connector line

Learn more

- [Information Flow](#) ^[1225]

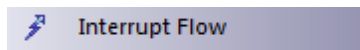
6.5.16 Interrupt Flow



Description

The *Interrupt Flow* is a connection used to define the two UML concepts of connectors for Exception Handler and Interruptible Activity Region. An Interrupt Flow is a type of activity edge. It is typically used in an Activity diagram.

Toolbox icon



Learn more

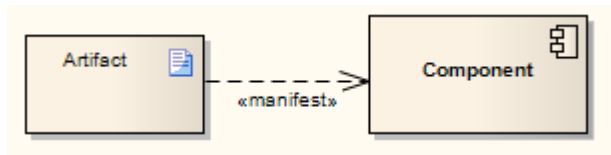
- [Exception Handler](#) ^[1123]
- [Interruptible Activity Region](#) ^[1147]
- [Activity diagram](#) ^[1028]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 327*) states:

An activity edge is an abstract class for directed connections between two activity nodes.

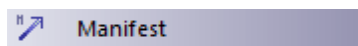
6.5.17 Manifest



Description

A Manifest relationship indicates that the Artifact source embodies the target model element, typically in Component and Deployment diagrams. Stereotypes can be added to Enterprise Architect to classify the type of manifestation of the model element.

Toolbox icon



Learn more

- [Artifact Element](#)^[1179]
- [Component Diagram](#)^[1023]
- [Deployment Diagram](#)^[1020]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 212*) states:

An artifact embodies or manifests a number of model elements. The artifact owns the manifestations, each representing the utilization of a packageable element.

Specific profiles are expected to stereotype the manifestation relationship to indicate particular forms of manifestation, e.g. «tool generated» and «custom code» might be two manifestations for different classes embodied in an artifact.

6.5.18 Message

Messages indicate a flow of information or transition of control between elements. Messages can be used by Timing Diagrams, Sequence Diagrams and Communication Diagrams (but not Interaction Overview diagrams) to reflect system behavior. If between Classes or classifier instances, the associated list of operations is available to specify the event.

Moving a Message can disrupt the organization of other features on the diagram. To avoid this, and move *only* the Message, press (**Alt**) while you move the Message.

Toolbox icon

⇒ Message**Learn more**

- [Message \(Sequence Diagram\)](#) ^[1232]
- [Message \(Communication Diagram\)](#) ^[1242]
- [Message \(Timing Diagram\)](#) ^[1245]
- [Timing Diagram](#) ^[1245]
- [Sequence Diagram](#) ^[1232]
- [Communication Diagram](#) ^[1242]

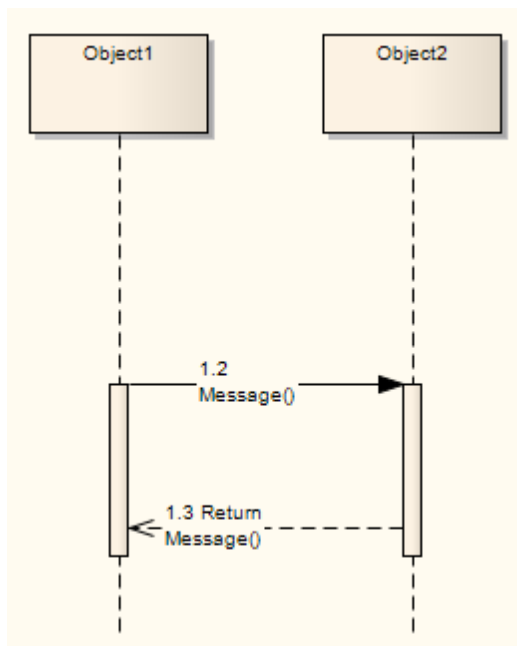
OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 491*) states:

A Message defines a particular communication between Lifelines of an Interaction.

A Message is a NamedElement that defines one specific kind of communication in an Interaction. A communication can be, for example, raising a signal, invoking an Operation, creating or destroying an Instance. The Message specifies not only the kind of communication given by the dispatching ExecutionSpecification, but also the sender and the receiver.

A Message associates normally two OccurrenceSpecifications - one sending OccurrenceSpecification and one receiving OccurrenceSpecification.

6.5.18.1 Message (Sequence Diagram)

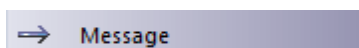
Sequence diagrams depict work flow or activity over time using Messages passed from element to element. These Messages correspond in the software model to Class operations and behavior.

How to

To create a Message on a Sequence diagram

Step	Action	See also
1	<p>Access the Sequence diagram.</p> <p>The Interaction pages of the Toolbox display.</p>	
2	<p>In the Interaction Relationships page, click on the Message icon, click on the source object and drag the cursor to the destination (target) object.</p> <p>The Message Properties dialog displays (if not, right-click on the Message and select the Message Properties context menu option).</p>	
3	<p>In the Message field, type the Message name:</p> <ul style="list-style-type: none"> • If the Message flow is <i>towards</i> a Class element (dropped in from a Class diagram) or a Lifeline element having a classifier, and the <i>destination</i> Class has defined <i>operations</i>, you can click on the drop-down arrow and select an appropriate operation name; the Message then reflects the destination Class operations • If the available operations are not appropriate, you can click on the Operations button and define a new operation in the target element, using the Operations dialog • If you create a Message without making reference to the target Class operations, no new operation is added to the target Class • 	<p>Class Element ^[1181]</p> <p>Lifeline ^[1144]</p> <p>Operations Dialog ^[901]</p>
4	<p>In the Parameters field, type any parameters that the Message has, as a comma-separated list.</p> <p>If required, in the Parameter Values field type the actual value for each parameter, again as a comma-separated list.</p>	
5	<p>If the Message is a return message, in the Return Value field enter the returned value or type.</p> <p>It is possible to depict returns from a Self Message; simply create a second Self Message at the end of execution and select the Is Return checkbox in the Control Flow Type panel.</p>	<p>Self-Message ^[1235]</p>
6	<p>If the Message flow is <i>from</i> a Class element or Lifeline element with classifier that has defined <i>attributes</i>, click on the drop-down arrow in the Assign to field and select an appropriate attribute name.</p> <p>The Message reflects the attributes from the <i>source</i> Class; you cannot add further attributes to the source Class here - if no appropriate attribute is listed, open the element Properties dialog and add the required attribute.</p> <p>Otherwise, if required, type the name of the object to assign the message flow to.</p>	

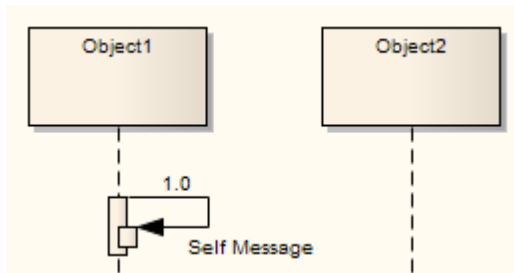
Step	Action	See also
7	In the Stereotype field, type or select an optional stereotype for the connector (this is displayed on the diagram, if entered).	
8	If required, in the Alias field type an alias for the name of the Message. On the diagram, the alias displays if the Use Alias if Available checkbox is selected on the Diagram tab of the Diagram Properties dialog. The Alias displays instead of or as well as the Message name, depending on the setting selected in the Alias Usage panel of the Diagram Behavior page of the Options dialog.	Diagram Behavior Options ^[553]
9	In the Condition field, type any conditions that must be true in order for the message to be sent.	
10	In the Synch: field in the Control Flow Type panel, select Synchronous or Asynchronous as appropriate.	Asynchronous Signal Message ^[1241]
11	In the Lifecycle field, select New to create a new element at the end of the Message, or Delete to terminate the message flow at the end of the Message. If neither case applies, leave the field at the default of <none> .	
12	If required, in the Notes field type any explanatory notes. You can format the notes using the Notes toolbar at the top of the field.	
13	Click on the OK button to save the Message definition. <ul style="list-style-type: none"> You can change the timing details of a message on the Timing Details dialog, and emphasize the sequence of closely-ordered messages using <i>General Ordering</i> To toggle the numbering of messages on a Sequence diagram, select or deselect the Show Sequence Numbering checkbox on the Options dialog 	Change the Timing Details ^[1238] General Ordering ^[1240] Sequence diagrams ^[1074]

Toolbox icon**Learn more**

- [Call](#) ^[1236]
- [Message Examples](#) ^[1237]

- [Co-Region Notation](#)^[1242]

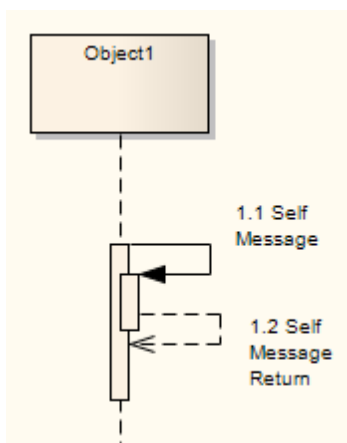
6.5.18.1.1 Self-Message



A Self-Message reflects a new process or method invoked within the calling lifeline's operation. It is a specification of a Message, typically in a Sequence diagram.

Self-Message as Return:

It is possible to depict a return from a Self Message call.



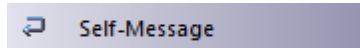
How to

To create a Self Message return

Step	Action
1	Create a second Self Message at the end of execution.
2	Double-click on the Message name to open the Message Properties dialog.
3	Select the Is Return checkbox.
4	Raise the Activation level of the return.

Self-Message Calls indicate a nested invocation; new activation levels are added with each Call.

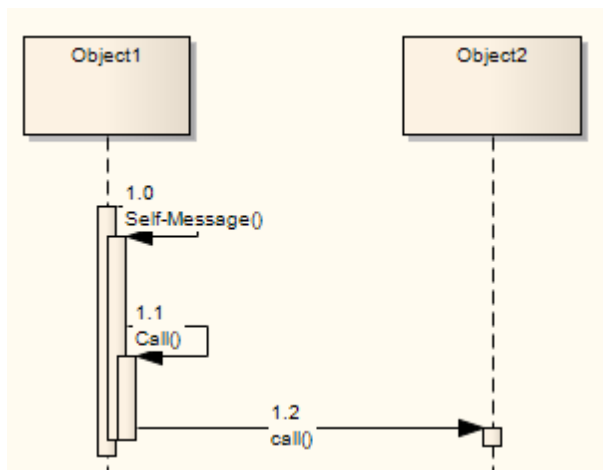
Toolbox icon



Learn more

- [Sequence Diagram](#)^[1074]
- [Message](#)^[1231]
- [Raise the Activation Level](#)^[1082]
- [Call](#)^[1236]

6.5.18.1.2 Call



A Call is a type of Message connector that extends the level of activation from the previous Message. All Self-Message Calls create a new activation level, but this focus of control usually ends with the next Message (unless activation levels are manually adjusted). Self-Message Calls, as depicted above by the first Call, indicate a nested invocation; new activation levels are added with each Call. Unlike a regular Message between elements, a Call between elements continues the existing activation in the source element, implying that the Call was initiated within the previous Message's activation scope.

Toolbox icon

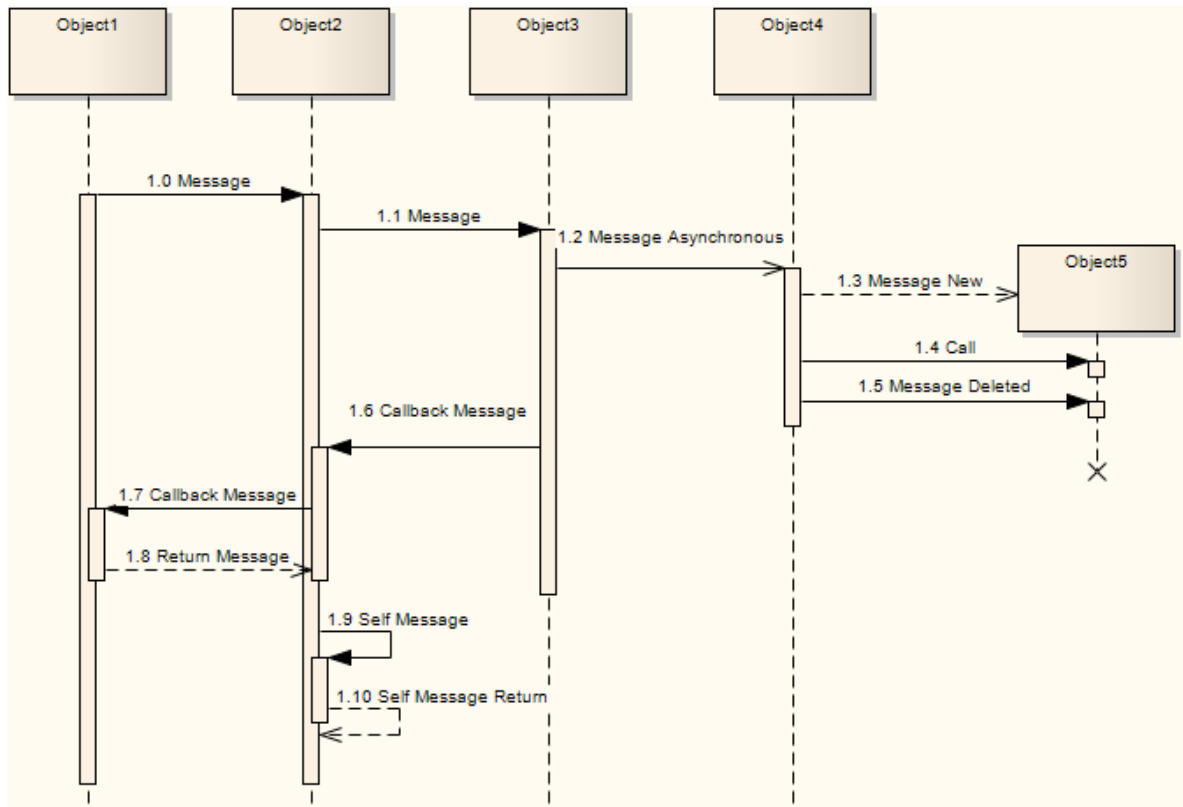


Learn more

- [Sequence Message](#)^[1232]
- [Self-Message](#)^[1235]
- [Sequence Element Activation](#)^[1080]

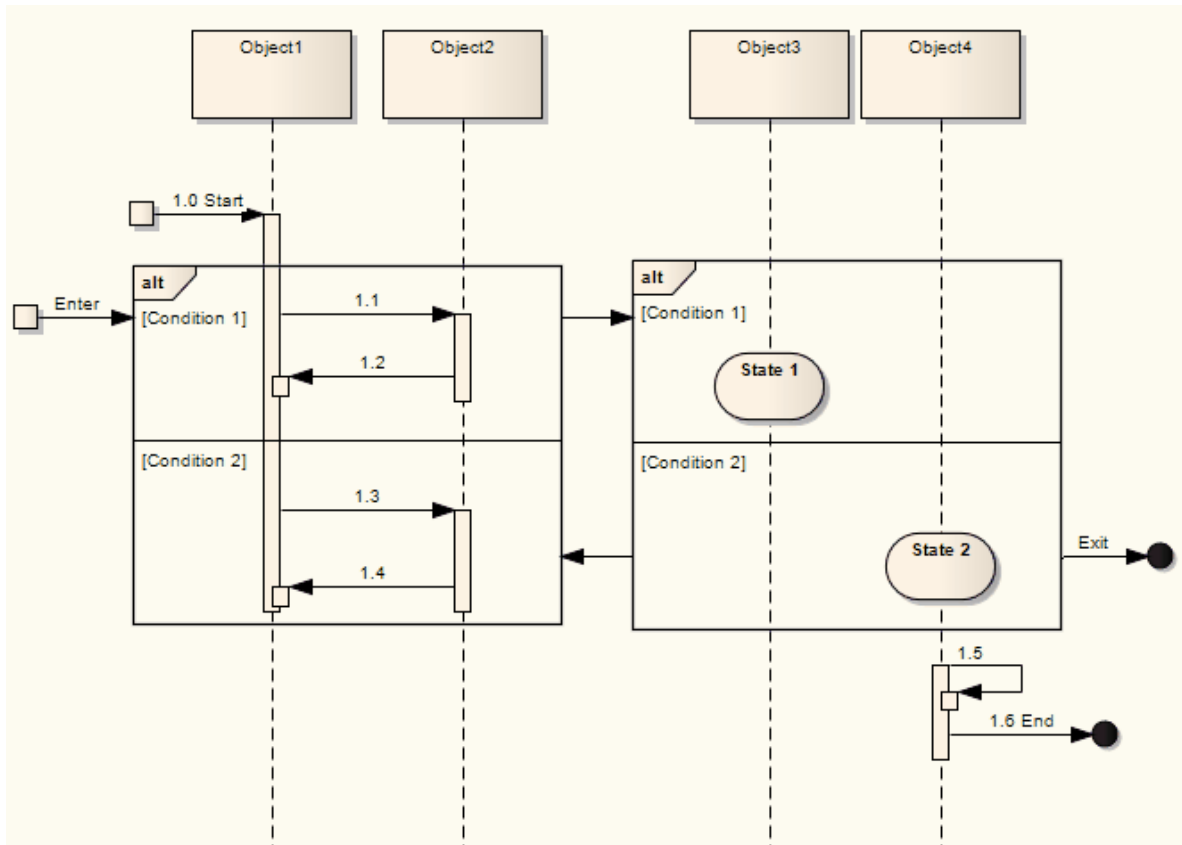
6.5.18.1.3 Message Examples

The following are different types of Messages available on Sequence Diagrams. Note that Messages on Sequence diagrams can also be modified with Shape Scripts.



Other Sequence Messages:

The following are examples of Messages that are not part of the sequence described by the diagram.



Learn more

- [Messages](#) ^[1232]
- [Sequence Diagrams](#) ^[1074]
- [Shape Scripts](#) ^[1358]

6.5.18.1.4 Change the Timing Details

It is possible to change the timing details of a Message in a Sequence diagram - right-click on the Message connector and select the **Timing Details** context menu option. The Timing Details dialog displays.

Complete the fields as indicated below:

Field	Action
Duration Constraint	Indicate the minimum and maximum limits on how long a message can last.
Duration Constraint Between Messages	Indicate the minimum and maximum interval between sending or receipt of the previous message at the current message's source Lifeline, and sending the current message.
Duration Observation	Capture the duration of a message.

Field	Action
Timing Constraint	Indicate the minimum and maximum time at which the message should arrive at the target.
Timing Observation	Capture the point at which the message was sent.

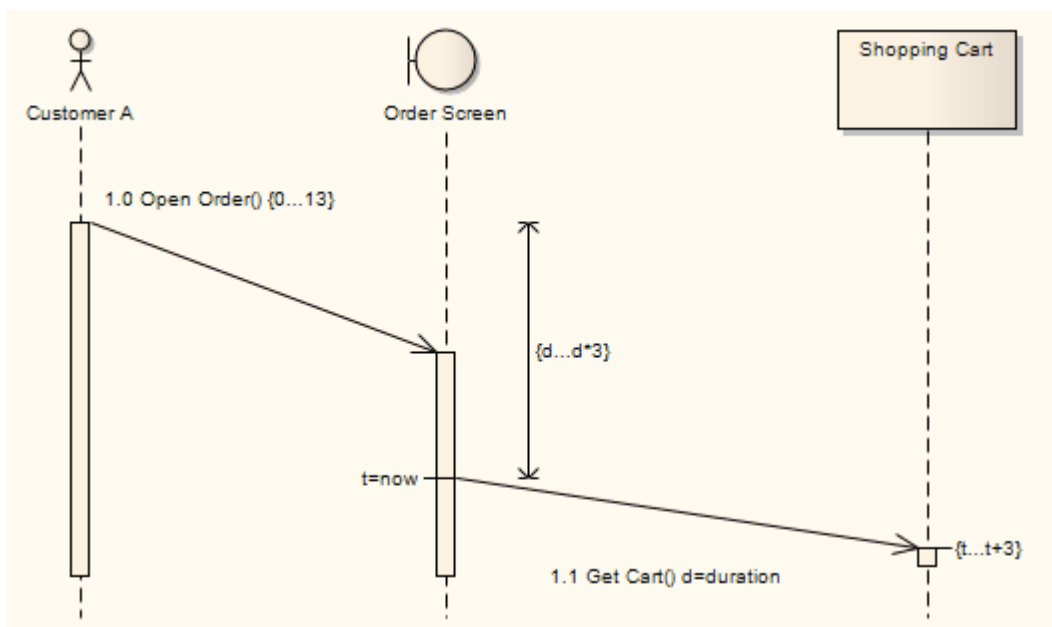
See the OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 511*).

In the diagram below, on the *Open Order* Message:

- **Duration Constraint** has been set to **0...13**

On the *Get Cart* Message:

- **Duration Constraint Between Messages** has been set to **d...d*3**
- **Duration Observation** has been set to **d=duration**
- **Timing Constraint** has been set to **t...t+3**
- **Timing Observation** has been set to **t=now**



By typing a value in the **Duration Constraint** field, you enable the Message angle to be adjusted. After clicking on the **OK** button on the Timing Details dialog, click on the head of the Message connector and drag the connector up or down to change the angle. You cannot extend the angle beyond the life line of the connecting sequence object or create an angle of less than 5 degrees.

You can also create the **Duration Constraint Between Messages** line by dragging the General Ordering

arrow up to the point at which the previous message joins the source Lifeline for the current message. A dialog displays on which you enter the value for the constraint. Having created the line, you can move it to any point within half way along the current message and half way along the previous message, to avoid overlap with other message timing details. You can edit or delete the value either **through the Timing Details** dialog or by right-clicking on the line itself and selecting the appropriate context menu option.

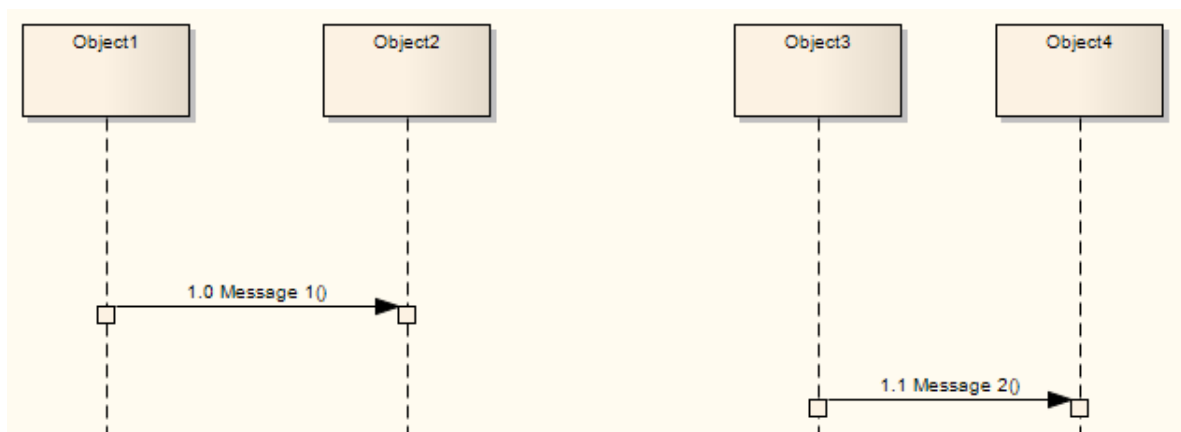
Learn more

- [Sequence Diagram](#)^[1074]
- [Message](#)^[1231]
- [General Ordering](#)^[1240]

6.5.18.1.5 General Ordering

In a Sequence diagram, the workflow is represented by the sequence of Messages down the diagram. Messages near the top of the diagram are passed before Messages lower down the diagram.

Consider the following diagram.

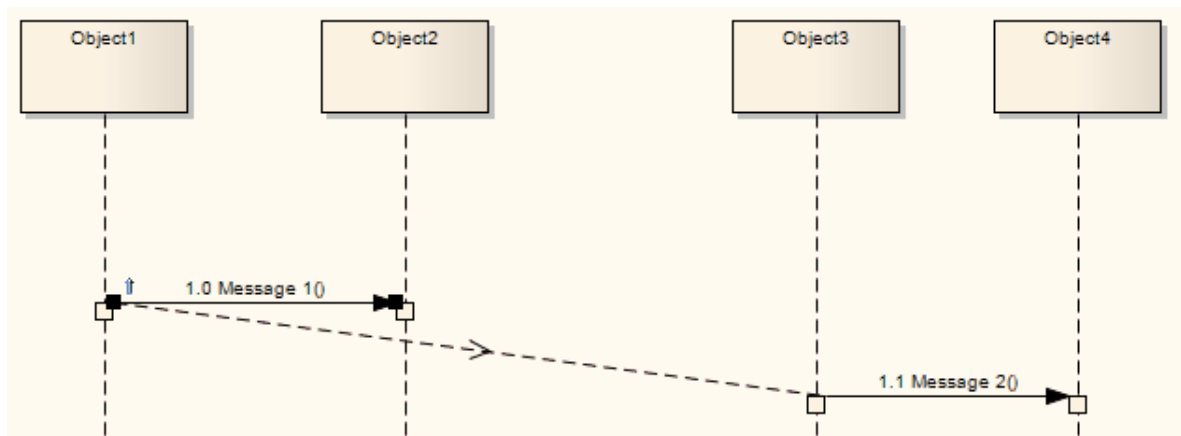


Message 1 is earlier than *Message 2*. However, in a complex diagram, or when representing finely timed operations or parallel processing, this might not be apparent. You can reinforce the sequence using a General Ordering arrow.

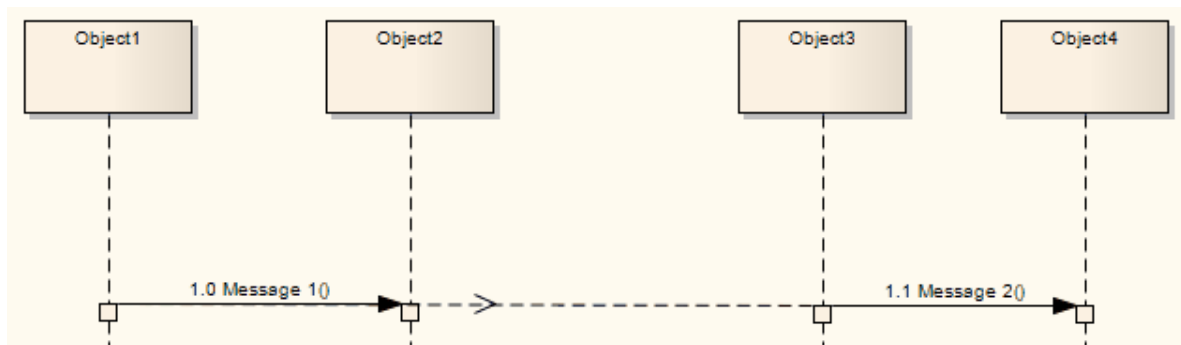
Click on the Message arrow. A small arrow displays at the source anchor point.



Click on this arrow and drag it to the start of the next Message in sequence (*Message 2* in the example). The General Ordering arrow displays, indicating that the second Message follows the first.



The General Ordering arrow is exaggerated in the above figure. You would normally have the arrow running almost horizontal across the diagram.



You can have more than one General Ordering arrow issuing from or targeting a Message, if necessary.

6.5.18.1.5.1 Asynchronous Signal Message

You define a Message as an asynchronous signal message by displaying the Message Properties dialog and setting the **Synch** field to **Asynchronous** and the **Kind** field to **Signal**. A *synchronous* message cannot be used to convey signals, so setting the **Synch** field to **Synchronous** disables the **Kind** field.

Return Value, **Assign To** and the **Operations** button, which are not applicable to asynchronous *signals*, are disabled.

The **Operations** button changes to a **Signal** button, which you click on to associate the asynchronous signal message with a Signal element in the model. You can type the arguments corresponding to the Signal attributes into the **Argument(s)** field.

When you click on the **Signal** button, the Select *Signal dialog displays, through which you locate and select the required Signal element. (The Select *Signal dialog is a variation of the Select <Item> dialog.)

Learn more

- [Message Properties Dialog](#)¹²³²

6.5.18.1.6 Co-Region Notation

Co-Region notation can be used as a short hand for parallel combined fragments. To access the **Co-Region** submenu, right-click on a connector in a Sequence diagram and select the **Co-Region** context menu option. There are four sub-options available:

- **Start at head**
- **End at head**
- **Start at tail**
- **End at tail**

6.5.18.2 Message (Communication Diagram)

A Message in a Communication diagram is equivalent in meaning to a Message in a Sequence diagram. It implies that one object uses the services of another object, or sends a message to that object. Communication Messages in Enterprise Architect are always associated with an Association connector between object instances. Always create the Association first, then add a Message to the connector.

Messages can be dragged into a suitable position by clicking and dragging on the message text.

Communication Messages are ordered to reflect the sequencing of the diagram. The numbering scheme should reflect the nesting of each event. A sequencing scheme could be:

```
1
2, 2.1, 2.2, 2.3
3.
```

This would indicate the single sequence of events 2.1, 2.2 and 2.3 occurs within an operation initiated by event 2. This is the default pattern applied by Enterprise Architect.

Alternatively, the sequence could be:

```
1
2, 2.1, 2.1.1, 2.1.1.1
  2.2, 2.2.1, 2.2.1.1
3
```

This would indicate that two sequences of events can be initiated by event 2, and 2.1 and 2.2 are separate sequences, not consecutive events in one sequence. You can set the sequence pattern and order using the Message Properties dialog and the Sequence Communications dialog.

If the target object is a Class or has its instance classifier set, the drop-down list of possible message names includes the exposed operations for the base type.

Learn more

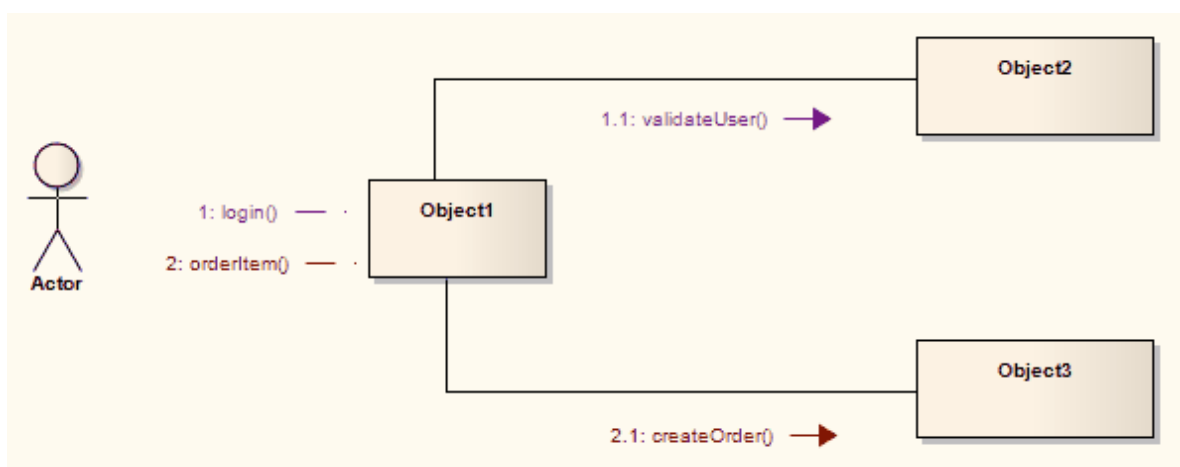
- [Communication Diagram](#) ^[1084]
- [Create a Communication Message](#) ^[1243]
- [Re-Order Messages](#) ^[1244]

6.5.18.2.1 Create a Communication Message

How to

To create a Communication Message

Step	Action
1	Open a diagram (one of: Communication, Analysis, Interaction Overview, Object, Activity or State Machine).
2	Add the required objects.
3	Add an Association relationship between each pair of objects that communicate.
4	Right-click on an Associate to display the context menu.
5	Select the option to add a Message from one object to the other.
6	When the Message Properties dialog displays, type in a name and any other required details.
7	Click on the OK button. The Message is added, connected to the Association and Object instances.
8	Move the Message to the required position.



Learn more

- [Communication Message](#)^[1242]
- [Message Properties Dialog](#)^[1232]

6.5.18.2.2 Re-Order Messages

When constructing your Communication diagram, it is frequently necessary to create or delete Message 'groups' and to re-order the sequence of Messages. There are two dialogs that help you perform these tasks: the Message Properties dialog and the Sequence Communications dialog.

Organize Message Groups:

If you have several Messages in the form 1.1, 1.2, 1.3, 1.4, for example, but would like to start a new numbering group on, say, the third Message (that is, 1.1, 1.2, **2.1**, 2.2, 2.3), you can change a Message in the series to a *Start Group* message.

How to

To reorganize message groups

Step	Action
1	Double-click on a Message <i>name</i> . The Message Properties dialog displays.
2	To make the selected Message the start of a new group, select the Start New Group checkbox.
3	If required, in the Notes field, type an explanatory note. You can format the text using the Notes toolbar at the top of the field.
4	Click on the OK button to save changes.

Sequence Messages:

In larger and more complex diagrams, you might have to use deeper levels of Messages in a group; for example, 1, 1.2, 1.2.1, 1.2.1.1. You might also have to change the sequence of Messages, making Message 1.3, for example, into Message 1.1.

How to

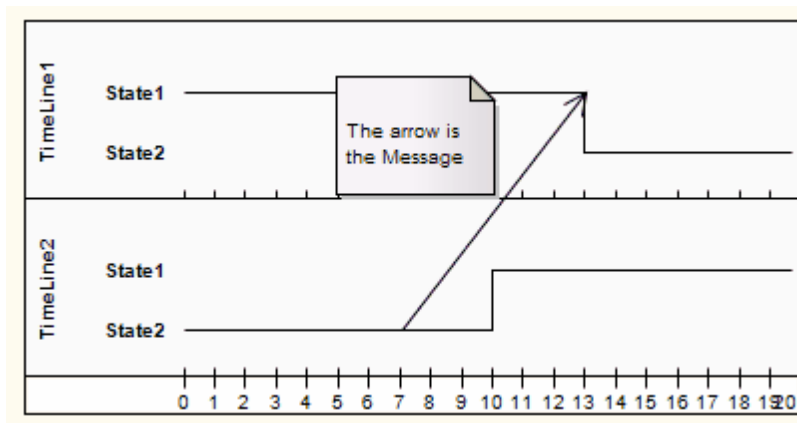
To change the sequence or level of Messages

Step	Action
1	<p>Either:</p> <ul style="list-style-type: none"> • Select the Diagram Advanced Sequence Messages menu option • Click on the diagram background and select the Sequence Communication Messages context menu option or • Right-click on a Message and select the Sequence Communication Messages context menu option <p>The Communication Messages dialog displays.</p>
2	<p>Click on the Message to adjust and, at the bottom of the dialog, click on the:</p> <ul style="list-style-type: none"> • Move Up or Move Down (Hand) buttons to move the Message up or down the sequence (e.g. Message 1.2 to Message 1.1 or 1.3) • Move Left or Move Right (Hand) buttons to move the Message up or down a level (e.g. Message 1.2.1 to Message 1.2 or Message 1.2.1.1)
3	<p>Repeat step 2 until the Message sequence and levels match your requirements. You might have to adjust other Message numbers (in group, sequence or level) to accommodate the changes you have made.</p>
4	<p>Click on the OK button to save changes.</p>

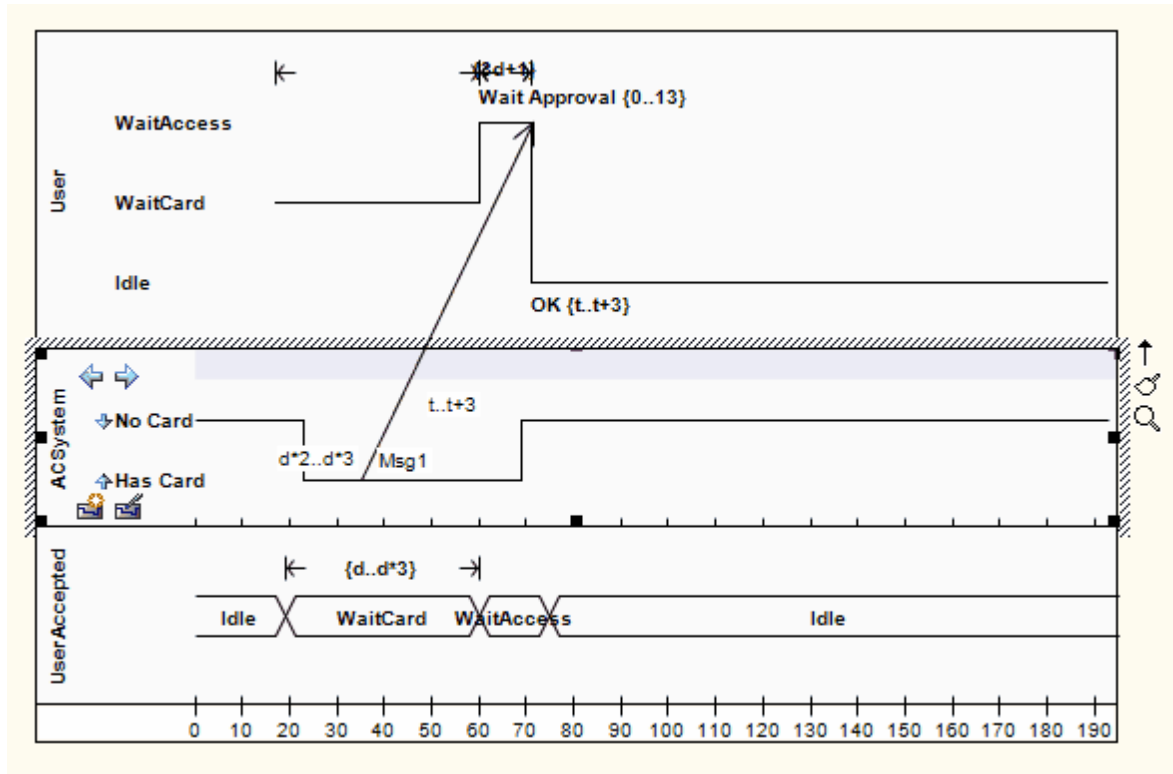
Learn more

- [Communication Message](#)^[1242]

6.5.18.3 Message (Timing Diagram)

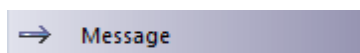


Messages are the communication links between Lifelines in a Timing diagram. In the case of a Timeline, a Message is a connection between two Timeline objects.



See *UML Superstructure Specification, v2.1.1, figures 14.30 and 14.31, p. 520.*

Toolbox icon



Learn more

- [Timing Diagram](#) ^[105]
- [Lifeline](#) ^[144]
- [Create a Timing Message](#) ^[1246]

6.5.18.3.1 Create a Timing Message

To create a Message in a Timing diagram, at least two Lifeline objects (State or Value) must be created first, each with existing transition points.

How to

To create a Message between Lifelines

Step	Action
1	Click on one of the Lifelines in the Timing diagram.
2	Select the Message icon from the Timing Relationships page of the Toolbox (More tools Timing).
3	Drag the cursor onto the Lifeline at the point at which the Message originates. The Timing Message dialog displays (if not, double-click on the Message).
4	The dialog consists of a set of transition points. Each transition point can be defined with the properties listed in the following table.

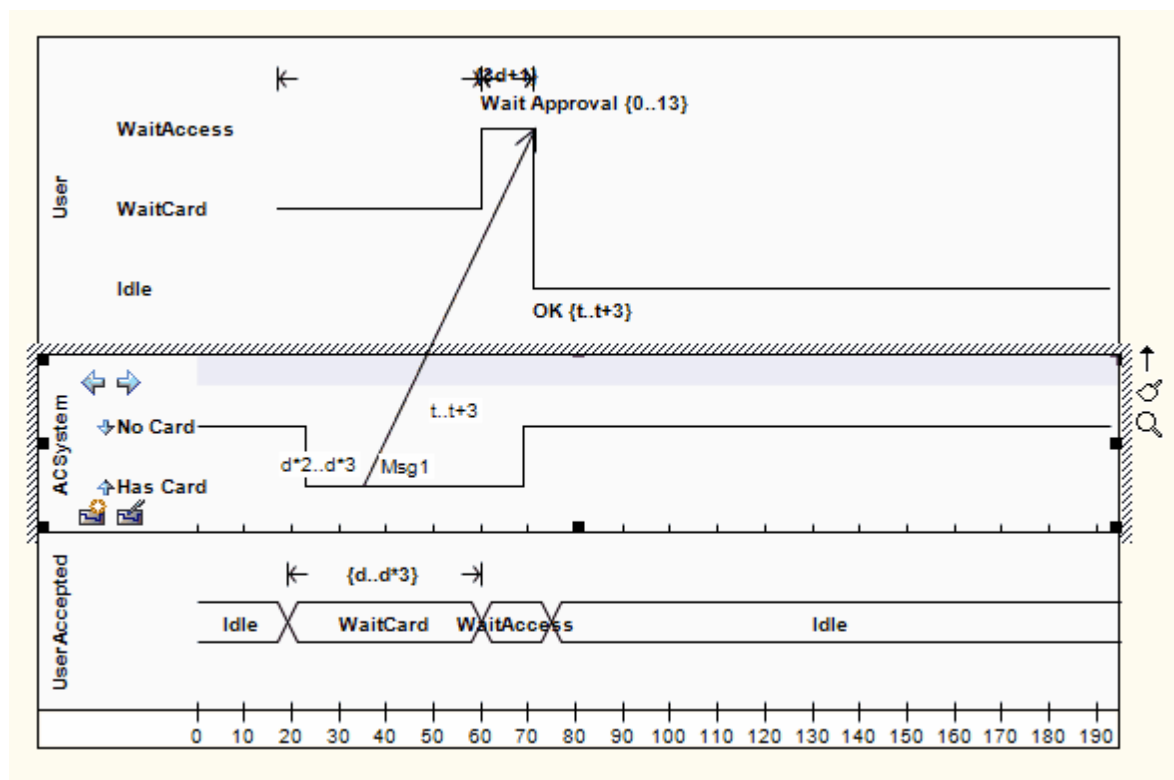
Property	Description
Start	Defines the lifeline where the message originates.
End	Defines the lifeline where the message terminates.

The following properties are set by default when a Message is created by dragging the cursor between two Lifelines.

Property	Description
Start Time	Specifies the start time for a message.
End Time	Specifies the end time for a message.
Name	The name of the message.
Time Observation	Provides information on the time of a sent message.
Duration Observation	Indicates the interval of a Lifeline at a particular state, begun from a message receipt.
Transition To	The state in the target Lifeline that the Message points to.

Property	Description
Event	The occurring event.
Time Constraint	The time taken to transmit a message.
Duration Constraint	Pertains to a lifeline's period at a particular state. The constraint could be instigated by that Lifeline's receipt of a message.

The following diagram shows an example of a configured Message:



See UML Superstructure Specification, v2.1.1, figures 14.30 and 14.31, p. 520.

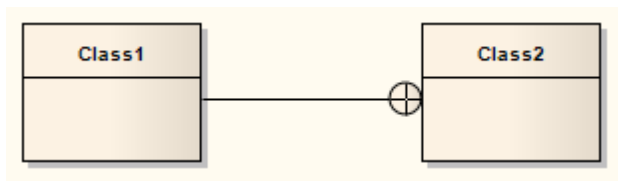
Notes

- You can move the source end of the Message freely along the source timeline; however, the target end (arrow head) must attach to a transition
- If you create a new Message and do not give it a target transition, it automatically finds and attaches to the nearest transition; if you move the target end, it drags the transition with it

Learn more

- [Timing Diagram](#) ^[1051]
- [Timing Message](#) ^[1245]
- [State Lifeline](#) ^[1158]
- [Value Lifeline](#) ^[1177]

6.5.19 Nesting



Description

The Nesting Connector is an alternative graphical notation for expressing containment or nesting of elements within other elements. It is most appropriately used for displaying Package nesting in a Package diagram.

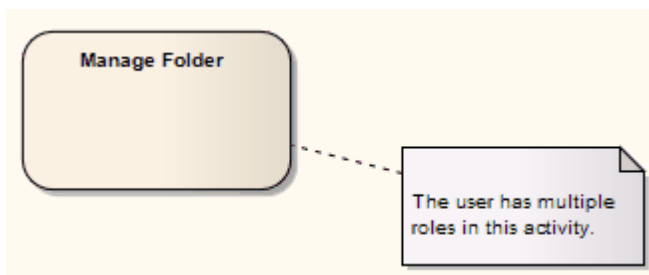
Toolbox icon



Learn more

- [Package Diagram](#) ^[1017]

6.5.20 Notelink

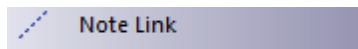


Description

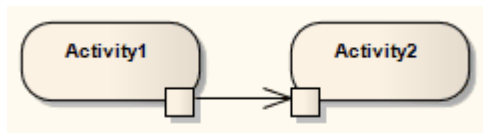
A Notelink connector connects a Note to one or more other elements of any other type.

Both Note and Notelink are available in any category of the Toolbox, in the Common page. You can also select them from the UML Elements toolbar.

Toolbox icon



6.5.21 Object Flow



Description

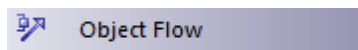
Object Flows are used in Activity diagrams and State Machine diagrams. When used in an Activity diagram, an Object Flow connects two elements, with specific data passing through it. To view sample Activity diagrams using Object Flows, see the *Object Flows in Activity Diagrams* topic.

In State Machine diagrams, an Object Flow is a specification of a state flow or transition. It implies the passing of an Object instance between elements at run-time.

You can insert an Object Flow from the State or Activity pages of the Toolbox, or from the drop-down list of all relationships located in the header toolbar. You can also modify a transition connection to an Object Flow by selecting the **ObjectFlow** checkbox on the connection Properties dialog.

See the *Control Flow* topic for information on setting up Guards and Weights on Object Flows.

Toolbox icon



Learn more

- [Activity Diagram](#) ^[1028]
- [Control Flow](#) ^[1219]
- [Object Flows in Activity Diagrams](#) ^[1250]

OMG UML Specification:

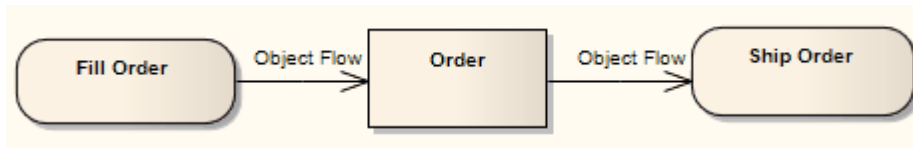
The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 389*) states:

An object flow is an activity edge that only passes object and data tokens.

6.5.21.1 Object Flows in Activity Diagrams

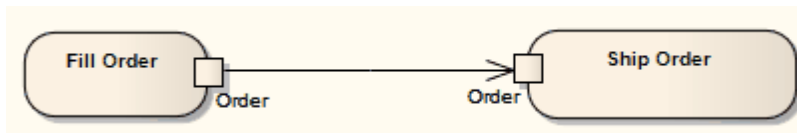
In Activity diagrams, there are several ways to define the flow of data between objects.

The following diagram depicts a simple Object Flow between two actions, *Fill Order* and *Ship Order*, both accessing order information.



See *UML Superstructure Specification, v2.1.1, figure 12.110, p. 391.*

This explicit portrayal of the data object *Order*, connected to the Activities by two Object Flows, can be refined by using the following format. Here, Action Pins are used to reflect the order.



See *UML Superstructure Specification, v2.1.1, figure 12.110, p. 391.*

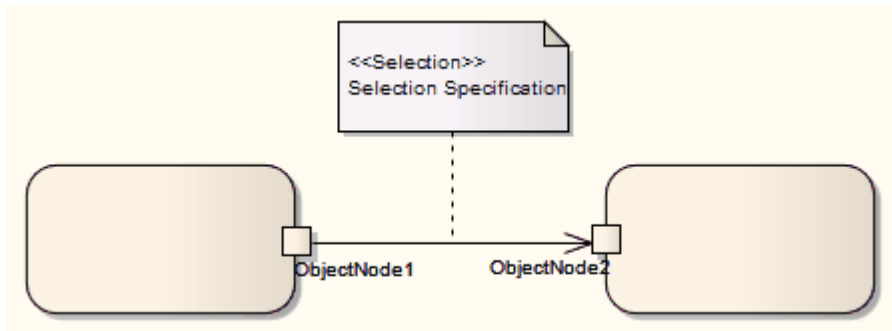
The following diagram is an example of multiple Object Flows exchanging data between two actions.



See *UML Superstructure Specification, v2.1.1, figure 12.111, p. 391.*

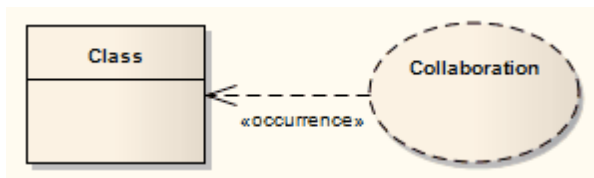
Selection and transformation behavior, together composing a sort of query, can specify the nature of the Object Flow's data access. Selection behavior determines which objects are affected by the connection. Transformation behavior might then further specify the value of an attribute pertaining to a selected object.

Selection and transformation behaviors can be defined by attaching a note to the Object Flow. To do this, right-click on the Object Flow and select the **Attach Note or Constraint** context menu option. A dialog lists other flows in the diagram, to which you can select to attach the note, if the behavior applies to multiple flows. To comply with UML 2, preface the behavior with the notation «*selection*» or «*transformation*».



See *UML Superstructure Specification, v2.1.1, figure 12.112, p. 392.*

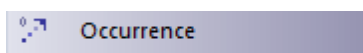
6.5.22 Occurrence



Description

An *Occurrence* relationship indicates that a Collaboration represents a classifier, in a Composite Structure diagram. An Occurrence connector is drawn from the Collaboration to the classifier.

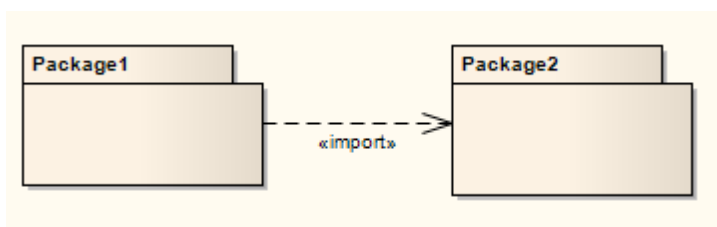
Toolbox icon



Learn more

- [Collaboration](#) ^[1184]
- [Composite Structure Diagram](#) ^[1017]

6.5.23 Package Import

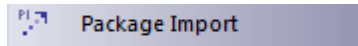


Description

A Package Import relationship is drawn from a source Package to a Package whose contents are to be

imported. Private members of a target Package cannot be imported. The relationship is typically used in a Package diagram.

Toolbox icon



Learn more

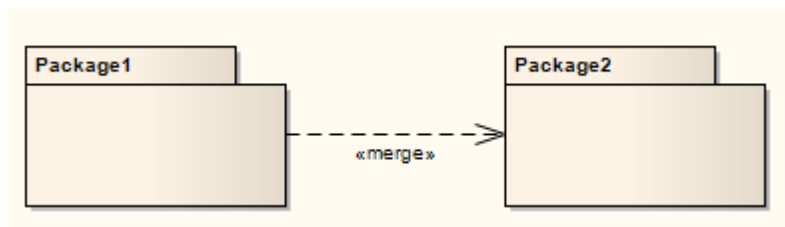
- [Package](#) ^[1200]
- [Package Diagram](#) ^[1017]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 112*) states:

A package import is a relationship between an importing namespace and a package, indicating that the importing namespace adds the names of the members of the package to its own namespace. Conceptually, a package import is equivalent to having an element import to each individual member of the imported namespace, unless there is already a separately-defined element import.

6.5.24 Package Merge



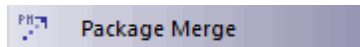
Description

In a Package diagram, a Package Merge indicates a relationship between two Packages whereby the contents of the target Package are merged with those of the source Package. Private contents of a target Package are not merged. The applicability of a Package Merge addresses any situation where multiple packages contain identically-named elements, representing the same thing. A Package Merge merges all matching elements across its merged Packages, along with their relationships and behaviors. Note that a Package Merge essentially performs generalizations and redefinitions of all matching elements, but the merged Packages and their independent element representations still exist and are not affected.

The Package Merge serves a graphical purpose in Enterprise Architect, but creates an ordered Package relationship applied to related Packages (which can be seen under the Link tab in the Package's Properties dialog). Such relationships can be reflected in XML exports or Enterprise Architect Automation Interface scripts for code generation or other Model Driven Architecture (MDA) interests.

Package Merge relationships are useful to reflect situations where existing architectures contain functionalities involving like elements, which are merged in a developing architecture. Merging doesn't affect the merged objects, and supports the common situation of product progression.

Toolbox icon



Learn more

- [Package Diagram](#)^[1017]
- [Package](#)^[1200]

OMG UML Specification:

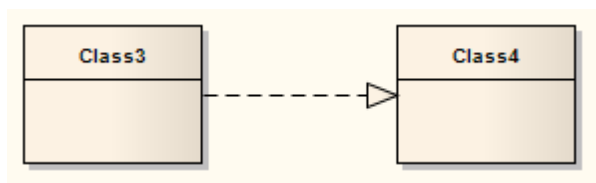
The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 113-114*) states:

A package merge is a directed relationship between two packages that indicates that the contents of the two packages are to be combined. It is very similar to Generalization in the sense that the source element conceptually adds the characteristics of the target element to its own characteristics resulting in an element that combines the characteristics of both.

This mechanism should be used when elements defined in different packages have the same name and are intended to represent the same concept. Most often it is used to provide different definitions of a given concept for different purposes, starting from a common base definition. A given base concept is extended in increments, with each increment defined in a separate merged package. By selecting which increments to merge, it is possible to obtain a custom definition of a concept for a specific end. Package merge is particularly useful in meta-modeling and is extensively used in the definition of the UML metamodel.

Conceptually, a package merge can be viewed as an operation that takes the contents of two packages and produces a new package that combines the contents of the packages involved in the merge. In terms of model semantics, there is no difference between a model with explicit package merges, and a model in which all the merges have been performed.

6.5.25 Realization



Description

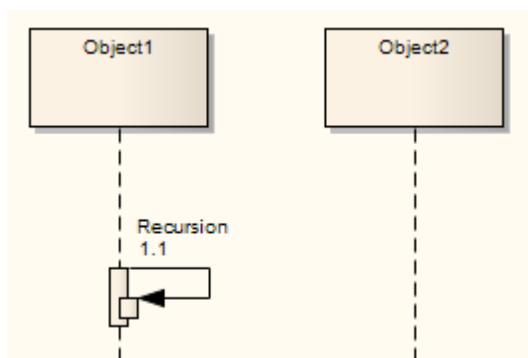
A source object implements or Realizes its destination object. Realize connectors are used in a Use Case, Component or Requirements diagram to express traceability and completeness in the model. A business process or Requirement is realized by one or more Use Cases, which in turn are realized by some Classes, which in turn are realized by a Component, and so on. Mapping Requirements, Classes and such across the design of your system, up through the levels of modeling abstraction, ensures the big picture of your system remembers and reflects all the little pictures and details that constrain and define it.

You can also define template binding parameters for a Realize connector between a binding Class and a parameterized Class.

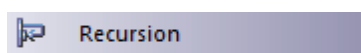
Toolbox iconOMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 131*) states:

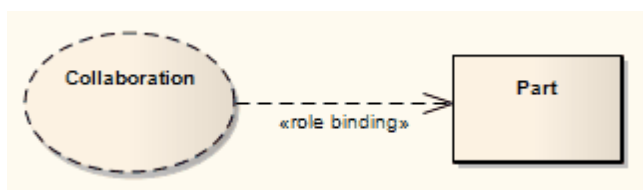
A Realization signifies that the client set of elements are an implementation of the supplier set, which serves as the specification. The meaning of 'implementation' is not strictly defined, but rather implies a more refined or elaborate form in respect to a certain modeling context. It is possible to specify a mapping between the specification and implementation elements, although it is not necessarily computable.

6.5.26 RecursionDescription

A Recursion is a type of Message used in Sequence diagrams to indicate a recursive function.

Toolbox iconLearn more

- [Message](#)^[1232]
- [Sequence Diagram](#)^[1074]

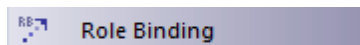
6.5.27 Role Binding

Description

Role Binding is the mapping between a Collaboration Use's internal roles and the respective Parts required to implement a specific situation, typically in a Composite Structure diagram. The associated Parts can have properties defined to enable the binding to occur, and the Collaboration to take place.

A Role Binding connector is drawn between a Collaboration and the classifier's fulfilling roles, with the Collaboration's internal binding roles labeled on the classifier end of the connector.

Toolbox icon



Learn more

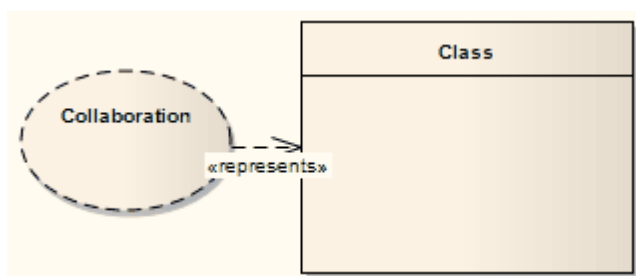
- [Collaboration Use](#) ^[1186]
- [Composite Structure Diagram](#) ^[1017]
- [Collaboration](#) ^[1184]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 174*) states:

A mapping between features of the collaboration type and features of the classifier or operation. This mapping indicates which connectable element of the classifier or operation plays which role(s) in the collaboration. A connectable element may be bound to multiple roles in the same collaboration use (that is, it may play multiple roles).

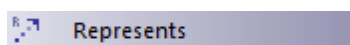
6.5.28 Represents



Description

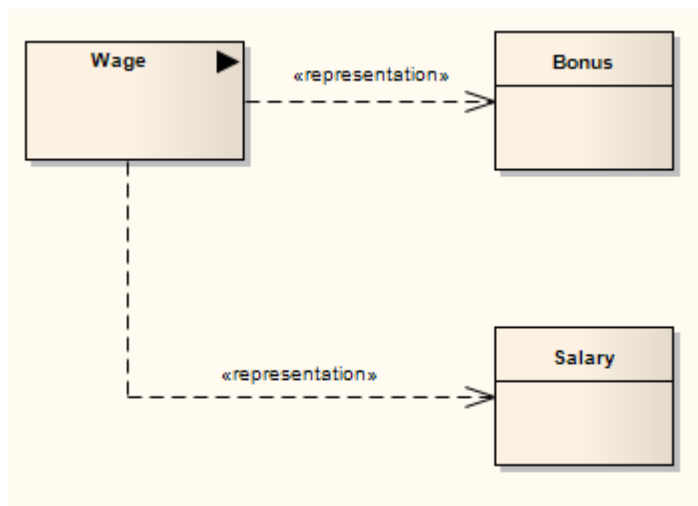
The Represents connector indicates that a Collaboration is used in a classifier, typically in a Composite Structure diagram. The connector is drawn from the Collaboration to its owning classifier.

Toolbox icon

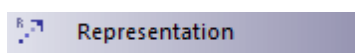


Learn more

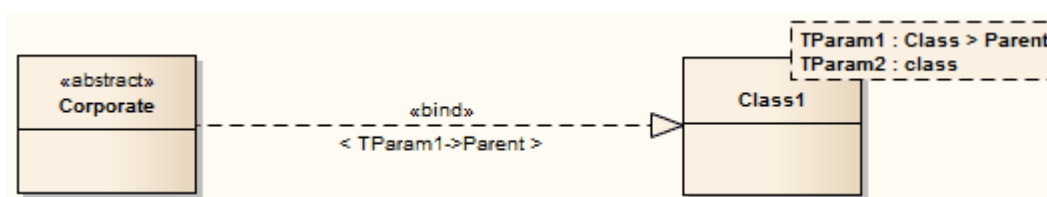
- [Collaboration](#)^[1184]
- [Composite Structure Diagram](#)^[1017]

6.5.29 RepresentationDescription

The Representation relationship is a specialization of a Dependency, connecting Information Item elements that represent the same idea across models, typically in an Analysis diagram. For example, Bonus and Salary are both a representation of the Information Item Wage.

Toolbox iconLearn more

- [Dependency](#)^[1221]
- [Information Item](#)^[1194]
- [Analysis diagram](#)^[1509]

6.5.30 Template Binding

Description

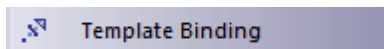
You create a Template Binding connector between a binding Class and a parameterized Class. You then define a binding expression on that connector. However, if the binding Class requires a Generalization, Realization or Association relationship with the parameterized Class, you can define the binding expression on that relationship instead.

You can create a Template Binding connector using:

- The **Template Binding** icon on the Class Relationships page of the Diagram Toolbox
- The Quicklinker arrow next to the source Class element
- The Templates tab of the binding Class element Properties dialog; here, you create the Template Binding relationship by clicking the **Add** button under the Binding(s) panel, specifying the connector type, and selecting the target parameterized Class from the Select <Item> dialog

Each of these methods creates the connector itself. For the first two methods you then double-click on the connector to display the connector Properties dialog, on which you select the Binding tab to define *parameter substitutions* as the binding expression. The third method takes you to this dialog and tab automatically.

Toolbox icon



Learn more

- [Parameterized Class](#) ^[1183]
- [Parameter Substitution](#) ^[1258]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 622*) states:

A template is a parameterized element ... used to generate other model elements using TemplateBinding relationships. The template parameters for the template signature specify the formal parameters that will be substituted by actual parameters (or the default) in a binding.

6.5.30.1 Parameter Substitution

Once a Template Binding (or other binding) relationship exists, you can add parameter substitutions to identify the formal parameters that are replaced, and the actual parameters that replace them, in the binding expression.

How to

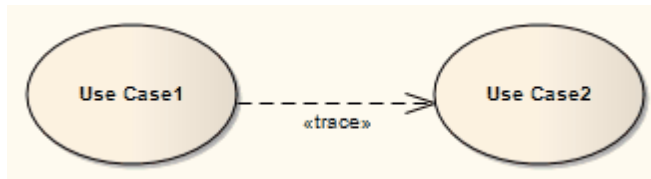
To define a parameter substitution

Step	Action
1	Display the Binding tab, in the Template Binding connector Properties dialog. The Target field identifies the target parameterized Class.
2	Click on the Add button below the Parameter Substitution(s) panel. The next available row in the panel is enabled for editing, and the word <i><none></i> is displayed in the Formal column.
3	Click on the field and on the drop-down arrow that is now displayed. A list of the template parameters from the target Class displays; click on the required parameter.
4	Click on the (...) button in the corresponding Actual field for the parameter. If the template parameter: <ul style="list-style-type: none"> • Does not have a constraint, a short context menu displays offering the choice of typing a free-text value into the Actual field, or selecting a classifier from the Select Classifier dialog • Has a constraint defined, the Select Classifier dialog displays automatically, showing the available classifiers
5	Locate and select the required classifier to replace the parameter in the binding expression. If you do not define an Actual classifier <i>and</i> the template parameter has a default value defined, that default is used in the expression.
6	To edit existing parameter substitutions, click on them and make the required changes as indicated in steps 3 and 4.
7	Click on the Apply and/or OK button. The parameter substitutions display as a label underneath the connector.

Learn more

- [Template Binding](#) 

6.5.31 Trace



Description

The Trace relationship is a specialization of a Dependency, connecting model elements or sets of elements that represent the same concept across models. Traces are often used to track requirements and model changes, typically in a Traceability diagram, or in a Class, Use Case, Object or Composite Structure diagram.

As changes can occur in both directions, the order of this Dependency is usually ignored. The relationship's properties can specify the trace mapping, but the trace is usually bi-directional, informal and rarely computable.

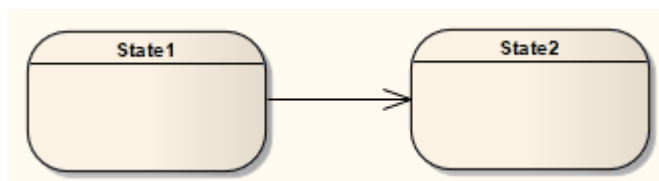
Toolbox icon



Learn more

- [Dependency](#) ^[1221]
- [Traceability Diagram](#) ^[649]

6.5.32 Transition



Description

A *Transition* defines the logical movement from one State to another, in a State Machine diagram. The Transition can be controlled through the following connector Properties dialog fields:

Field	Action	See also
Guard	Type in the expression that is to be evaluated after an Event is dispatched, but before the corresponding Transition is triggered. If the guard is true at that time, the Transition is enabled; otherwise, it is disabled.	

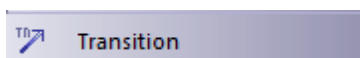
Field	Action	See also
Effect is a Behavior	<p>Convert the Effect field from a free-text field to the definition of a specific Activity or behavior.</p> <p>Enterprise Architect displays the Select <Item> dialog to prompt you to select the Activity or behavior element from the model.</p>	Select <Item> Dialog ^[880]
Effect	<p>Either:</p> <ul style="list-style-type: none"> • Type a description of the effect of the Transition, or • If you have selected the Effect is a Behavior check box, select an Activity or behavior to be performed during the Transition (to change this subsequently, click on the (...) button to redisplay the Select <Item> dialog) 	
Trigger		
Name	<p>Specify the name of the trigger; either:</p> <ul style="list-style-type: none"> • Type the name, or • Select an existing trigger in the model using the Select <Item> dialog, which you display by clicking on the (...) button 	
Type	<p>Specify the type of trigger: Call, Change, Signal or Time:</p> <ul style="list-style-type: none"> • Call - specifies that the event is a CallEvent, which sends a message to the associated object by invoking an operation • Change - specifies that the event is a ChangeEvent, which indicates that the transition is the result of a change in value of an attribute • Signal - specifies that the event is a SignalEvent, which corresponds to the receipt of an asynchronous signal instance • Time - corresponds to a TimeEvent; which specifies a moment in time <p>Code generation for State Machines currently supports Change and Time trigger events only, and expects a specification value.</p>	
Specification	<p>Specify the event instigating the Transition; either:</p> <ul style="list-style-type: none"> • Type the event (time or change), or 	

Field	Action	See also
	<ul style="list-style-type: none"> Select an existing specification in the model using the Select <Item> dialog, which you display by clicking on the (...) button 	
New	Clear the fields ready to begin defining a new trigger.	
Save	Save the newly created or edited trigger.	
Delete	Remove the selected trigger from the list.	
<trigger list>	List the existing triggers, which might or might not have names and types, and which can include Call and Signal triggers created in older models.	

Notes

- Fork and Join segments can have neither triggers nor guards
- You can identify hidden triggers and locate triggers in the Project Browser, using the **Find Triggers Associated** option on the Transition connector context menu - if one trigger exists for the Transition, it is immediately highlighted in the Project Browser; if more than one trigger exists, the Element Usage dialog displays; select the required trigger and click on the **Open** button to highlight the trigger in the Project Browser

Toolbox icon



Learn more

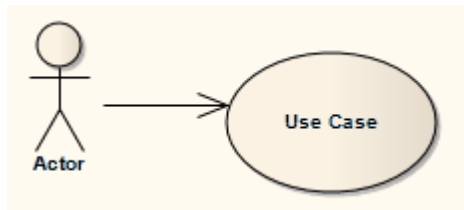
- [State Machine Diagrams](#) ^[1032]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 568*) states:

A transition is a directed relationship between a source vertex and a target vertex. It may be part of a compound transition, which takes the state machine from one state configuration to another, representing the complete response of the state machine to an occurrence of an event of a particular type.

6.5.33 Use



Description

A Use relationship indicates that one element requires another to perform some interaction. The Use (or Usage) relationship does not specify how the target supplier is used, other than that the source client uses it in definition or implementation. A Use relationship is a sub-typed Dependency relationship.

You typically use the Use relationship in Use Case diagrams to model how Actors use system functionality (Use Cases), or to illustrate usage dependencies between Classes or Components.

Notes

- It is more usual (and correct UML) to have an Association between an Actor and a Use Case
- To depict a usage dependency on a Class or Component diagram, draw a Dependency connector; right-click on the Dependency, and select the **Dependency Stereotypes | Use** context menu option

Toolbox icon



Learn more

- [Use Case Diagram](#)^[1030]
- [Association](#)^[1210]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 138*) states:

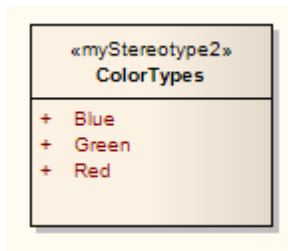
A usage is a relationship in which one element requires another element (or set of elements) for its full implementation or operation. In the metamodel, a Usage is a Dependency in which the client requires the presence of the supplier.

6.6 UML Stereotypes

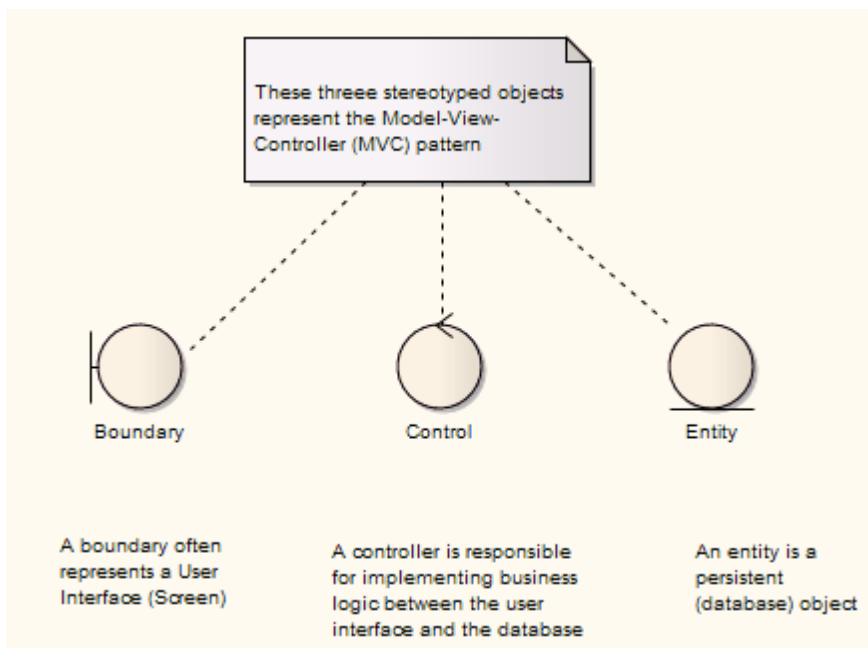
UML supports a large number of *stereotypes*, which are an inbuilt mechanism for logically extending or altering the meaning, display and syntax of a model element. Different model elements have different standard stereotypes associated with them.

For further definition of stereotypes, see the OMG UML specification (*UML Superstructure Specification, v2.1.1, section 18.3.8, pp. 667-672*).

A stereotype is generally displayed as in the example below (where «myStereotype2» is the stereotype).



In some cases the stereotype causes the element to be drawn differently, as below:



New, or customized, stereotypes can be created. Stereotypes can also be associated with new shapes, using either metafiles (image files) and colors or Shape Scripts, to apply non-UML shapes to elements and connectors. For further information on customizing stereotypes and applying Shape Scripts, see the *MDG Technology SDK* topic.

[Learn more](#)

- [Apply Stereotypes](#)^[1265]
- [Stereotype Selector](#)^[1266]
- [Stereotype Visibility](#)^[1266]
- [Standard Stereotypes](#)^[1267]
- [Stereotypes with Alternative Images](#)^[1269]
- [MDG Technology SDK](#)^[1292]
- [Customizing Stereotypes](#)^[1293]
- [Shape Scripts](#)^[1358]

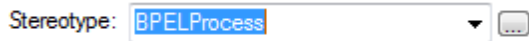
6.6.1 Apply Stereotypes

Enterprise Architect enables you to apply one or more stereotypes to any UML construct, including:

- Elements (such as Classes and Objects)
- Relationships (such as Dependencies and Associations)
- Association Ends
- Attributes and Operations
- Operation Parameters

How to

To apply a stereotype to any UML construct, using the Properties *dialog*, select any *one* of the following steps:

Step	Action	See also
1	In the Stereotype field, type the stereotype(s) to apply as a comma-separated list.	
2	Click on the drop-down arrow and select the required stereotype from the list.	
3	Click on the (...) button to use the Stereotype Selector dialog. 	Stereotype Selector ^[1266]

To apply a stereotype to an element using the Properties *window*, select any *one* of the following steps:

Step	Action	See also
1	In the Stereotype field, type the stereotype(s) to apply as a comma-separated list.	

Step	Action	See also
2	Click on the drop-down arrow and select the required stereotype from the list.	
3	Select the browse other stereotypes... option in the drop-down list to use the Stereotype Selector dialog.	Stereotype Selector ^[1266]

6.6.2 Stereotype Selector

The Stereotype Selector dialog enables you to apply one or more stereotypes to a UML construct, from multiple stereotype sources such as Profiles or the Custom Stereotypes list. The appearance of the stereotype is influenced by the stereotype visibility settings on the Diagram Properties dialog.

How to

To select stereotypes to apply/remove

Step	Action
1	On the element or connector Properties dialog, click on the (...) button near the Stereotype field. The Stereotype for: <object type> dialog displays.
2	Click on the Profile drop-down arrow and choose the required stereotype source.
3	In the Stereotypes list, enable or disable the required stereotype by selecting or deselecting the checkbox against it.
4	Click on the OK button to apply the selection.

You can also define a new stereotype to apply to the required construct by clicking on the **New...** button and entering the name of the new stereotype when prompted.

Learn more

- [Applying Stereotypes](#)^[1265]
- [Stereotype Visibility](#)^[1266]

6.6.3 Stereotype Visibility

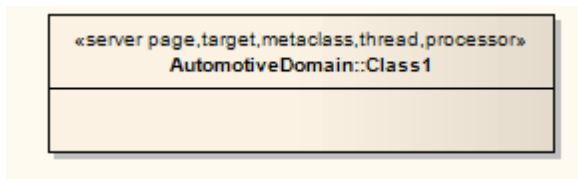
You control the visibility of applied stereotypes using three options in the diagram Properties dialog. Select:

- The **Show Element Stereotypes** checkbox (on the Elements page) to show or hide all element stereotypes in the current diagram

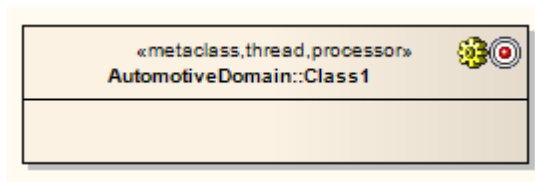
- The **Show Stereotypes** checkbox (for features, on the Features page) to show or hide all attribute and operation stereotypes in the current diagram
- The **Use Stereotype Icons** checkbox (on the Elements page) to display icons, instead of strings, for those stereotypes that have icons defined

The example below shows how a Class would appear having multiple stereotypes applied to it.

Use Stereotype Icons disabled: displays all the applied stereotypes in a comma-separated string within «guillemets».



Use Stereotype Icons enabled: displays icons for those stereotypes with icons defined. Stereotypes without icons defined are still displayed in the comma-separated string.



6.6.4 Standard Stereotypes

Below is a list of standard element stereotypes (as provided in the *EABase.eap* base model), each enclosed by guillemets (« »):

Stereotype	Base Class
«access»	Dependency
«become»	Flow
«call»	Usage
«copy»	Flow
«create»	Message
«derive»	Abstraction
«destroy»	Message

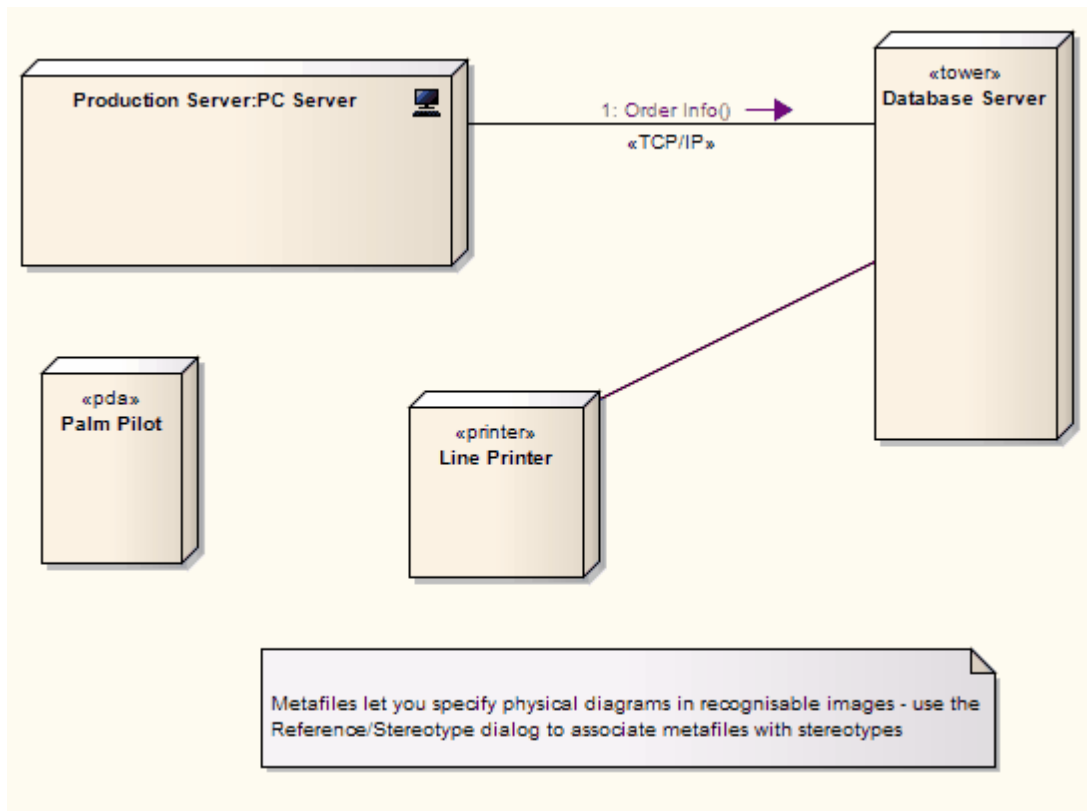
Stereotype	Base Class
«document»	Abstraction
«executable»	Abstraction
«facade»	Package
«file»	Abstraction
«framework»	Package
«friend»	Dependency
«global»	AssociationEnd
«implementation»	Class
«implementation»	Generalization
«import»	Dependency
«instantiate»	Usage
«invariant»	Constraint
«library»	Abstraction
«local»	AssociationEnd
«metaclass»	Class
«parameter»	AssociationEnd
«postcondition»	Constraint

Stereotype	Base Class
«powertype»	Class
«precondition»	Constraint
«process»	Classifier
«refine»	Abstraction
«requirement»	Comment
«responsibility»	Comment
«self»	AssociationEnd
«send»	Usage
«stub»	Package
«table»	Abstraction
«thread»	Classifier
«trace»	Abstraction
«type»	Class
«utility»	Classifier

6.6.5 Stereotypes with Alternative Images

You can alter the appearance of elements using stereotypes. This does not apply to elements that include Lifelines, such as those in Sequence diagrams.

If the stereotype has an associated metafile, when the stereotype is applied to a Class or other element that supports alternative graphical format, Enterprise Architect then draws the alternative image instead of the standard one.



6.7 Design Patterns

Design patterns are a group of collaborating Objects/Classes that can be abstracted from a general set of modeling scenarios. They are also known as parameterized collaborations.

Design patterns are an excellent means of achieving re-use and building in robustness. As patterns are discovered in any new project, the basic pattern template from previous engagements can be re-used with the appropriate variable names modified for the current project.

Patterns generally describe how to solve an abstract problem, and it is the task of the pattern user to modify the pattern elements to meet the demands of the current engagement.

Before using a pattern it must first be created as a standard UML diagram and then saved as an XML pattern file. This XML file can then be imported as a resource that can be used in any model.

Sparx-Created GoF Patterns:

To get you started with design patterns in Enterprise Architect, Sparx Systems provides you with an MDG technology for the patterns described in the book *Design Patterns - Elements of Reusable Object-Oriented Software* by Gamma et al., referred to as the 'Gang of Four' or GoF. These patterns are made available through a set of Toolbox pages.

The pattern elements are drawn from the *EABase.eap* file, through the Resources window hierarchy. Therefore, if you are developing your model in a DBMS repository (or you inadvertently delete the GoF patterns from your .eap file) you can download them as a 'zip' file from www.sparxsystems.com/uml_patterns.html.

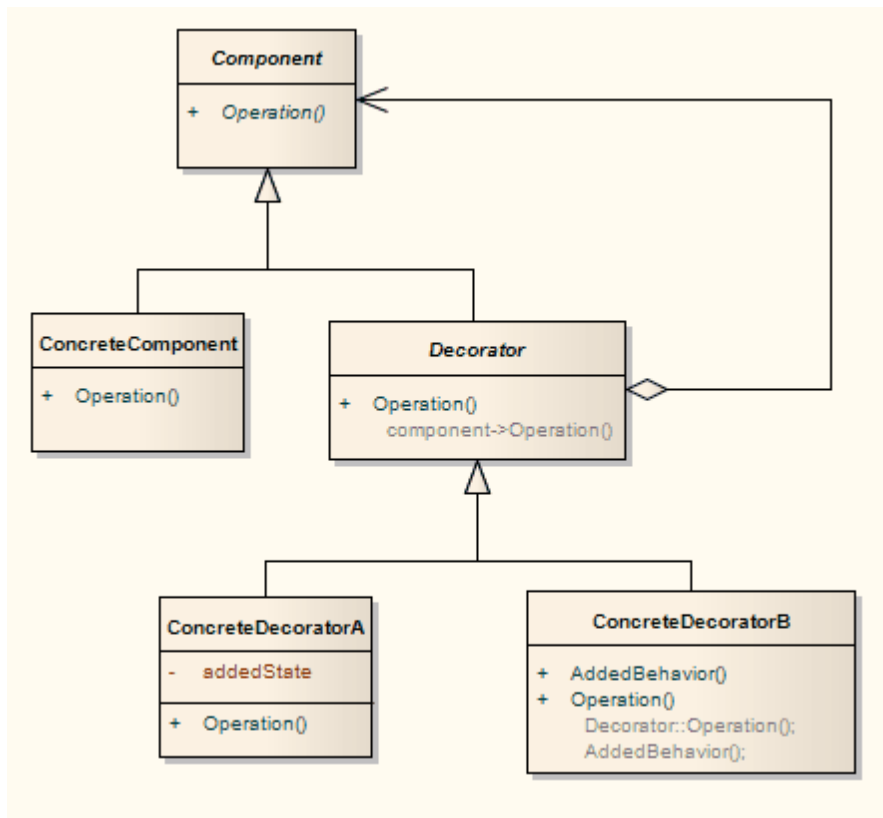
Because the patterns are drawn from the Resources window, if you delete a pattern in the Resources window the equivalent Toolbox item cannot work. Therefore, if you cannot drop a pattern element from the Toolbox, check that it is still available in the Resources window.

Learn more

- [GoF Patterns](#)^[1928]
- [Create a Pattern](#)^[1271]
- [Import a Pattern](#)^[1273]
- [Use a Pattern](#)^[1274]

6.7.1 Create a Pattern

To create a Design Pattern you first must model the Pattern as a standard UML diagram within Enterprise Architect. The following diagram was created from an example in the GoF book *Design Patterns - Elements of Reusable Object-Oriented Software* by Gamma et al.




[Access](#) [Select required diagram](#) | [Diagram](#) | [Advanced](#) | [Save UML Pattern](#)

Reference

On the Save Diagram as UML Pattern dialog, complete the fields as indicated below:

Field/Button	Action	See also
Pattern Name	Type the Pattern name.	
Filename	Type a directory path and .XML filename into which to save the Pattern.	
Category	Type the Category under which the Pattern should be listed in UML Patterns (required).	
Version	Type the Pattern version number.	
Notes	Type any notes on the Pattern.	
Actions	Select the appropriate checkboxes to select the actions for the	Use a Pattern ¹²⁷⁴

Field/Button	Action	See also
	<p>elements that are contained in the Pattern; these actions are performed when the Pattern is used.</p> <p>The available actions are:</p> <ul style="list-style-type: none"> • Create: Creates the Pattern element directly without modification • Merge: Merges the Pattern element with an existing element, enabling the existing element to take on the role of the selected Pattern element • Instance: Creates the Pattern element as an instance of an existing element • Type: Creates the Pattern element types as an existing element <p>If your Pattern includes an Object element, you would use Instance to set the classifier of the Object to one of the Classes in the diagram onto which you are dropping the Pattern.</p> <p>If your Pattern includes a Property (Port or Part) you would use Type to set the type of the Property to one of the Classes in the diagram onto which you are dropping the Pattern.</p>	
OK	<p>Click on this button twice to save the Pattern.</p> <p>Once saved you can load the Pattern into Enterprise Architect, into the Resources window.</p>	Import a Pattern 

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Diagrams** permission to save a diagram as a Pattern
- If your source diagram contains information flows, the Information Items Conveyed and Information Flows Realized data is not copied into the Pattern
- To change the name of one of the elements, double-click on the element to display the Edit dialog; from this dialog you can also add comments detailing the element's purpose

6.7.2 Import a Pattern

Before using a previously created Design Pattern file in a model, you must first import it into the model; it is then available from the Resources window and optionally from the Toolbox.

How to

To import a Design Pattern you have previously saved

Step	Action
1	Select the Resources window.
2	Right-click on the <i>UML Patterns</i> node and select the Import UML Pattern context menu option. The Select UML Pattern Import Filename dialog displays.
3	Locate the XML file to import.
4	Click on the Open button to import the Pattern. The imported Pattern is placed in the appropriate category as defined in the XML file; if the category does not already exist under <i>UML Patterns</i> , a new one is created.

Gang of Four patterns are integrated with Enterprise Architect in the *EABase.eap* file; however, if you create your model in a DBMS repository (or you inadvertently delete the patterns from your model .eap file) you can use the above procedure to download examples of the Gang of Four patterns from the GoF Patterns zip file on the Sparx Systems website.

Learn more

- [Create a Pattern](#)^[1277]
- [GoF Patterns zip file](#)

6.7.3 Use a Pattern

Using a Design Pattern enables you to rapidly create template solutions for code structures that perform the same type of task in other situations, and to use items defined in the Pattern with the UML model.

How to

To use a Pattern that you have previously imported into the model

Step	Action	See also
1	Open the diagram into which to add the Pattern.	
2	Select the Resources window.	
3	Expand the <i>UML Pattern</i> folder and find the Pattern to add.	
4	Either: <ul style="list-style-type: none"> • Right-click on the Pattern and select the Add Pattern to Diagram context 	Add Pattern Dialog ^[1276]

Step	Action	See also
	<p>menu option or</p> <ul style="list-style-type: none"> • Drag and drop the Pattern from the Resources window onto the diagram <p>You can also view the Pattern details in read-only mode by selecting the View Pattern Details context menu option.</p>	
5	Once the appropriate selections have been made, click on the OK button to import the Pattern into the model, recreating the original diagram with new GUIDs.	

To change the default of the Pattern element

Step	Action	See also
1	From the Add Pattern dialog select the individual element in the Pattern Element panel.	
2	Click on the (...) button to display the Edit dialog. The specific method for changing the element name is dependant upon the entry in the Action column of the Pattern Elements panel.	
3	If the Action entry for the element is Create , then in the Default field in the Edit dialog delete the existing value and type your own, user-defined value. Click on the OK button. The element default is updated on the Add Pattern dialog.	
4	If the Action entry for the element is Merge , in the Edit dialog click on the (...) button to browse to an existing element classifier. The Select <Item> dialog displays.	
5	Locate and select an existing element classifier. You can restrict the number of choices by selecting the elements from a specific namespace; to do this, click on the In Namespace drop-down arrow and select a namespace.	Using Classifiers ⁸⁹⁹

Learn more

- [Import a Pattern](#) ¹²⁷³

6.7.4 Add Pattern Dialog

The Add Pattern dialog displays when you are using or editing a Design Pattern element.

Reference

Panel	Action
Preview	Display a preview of the Pattern. Click on the Preview link to open a view of the Pattern and drag the sides into as large a picture as you require.
Pattern Elements	Access the individual elements contained in the Pattern. From here you can: <ul style="list-style-type: none">• Select the action for the individual element (<i>Create</i>, <i>Merge</i>, <i>Instance</i> or <i>Type</i>, as applicable for each element) by clicking on the drop-down arrow, or• Modify the default of the Pattern element or - for a merged element - choose the namespace, by clicking on the (...) button on the right of the Default entry
Element Notes	Display the comments that describe the element in the Pattern. Highlight an element in the Pattern Elements panel to view the notes.

6.8 UML Profiles

UML Profiles provide a means of extending UML, which enables you to build UML models in particular domains. They are based on additional stereotypes and Tagged Values that are applied to elements, attributes, methods, connectors, connector ends and so on. A Profile is a collection of such extensions that together describe some particular modeling problem and facilitate modeling constructs in that domain. For example, the UML Profile for XML describes a set of extensions to basic UML model elements to enable accurate modeling of XSD Schemas (see *Modeling XML Applications with UML*, David Carlson, p. 310). You can create your own Profiles to describe modeling scenarios specific to your development environment.

Enterprise Architect has a generic UML Profile mechanism for loading and working with different Profiles. UML Profiles for Enterprise Architect are specified in XML files, with a specific format. You can reference these XML files in Enterprise Architect as part of an MDG Technology (recommended) or import them through the Resources window (not recommended). The imported Profile automatically generates a page of elements and relationships in the Toolbox.

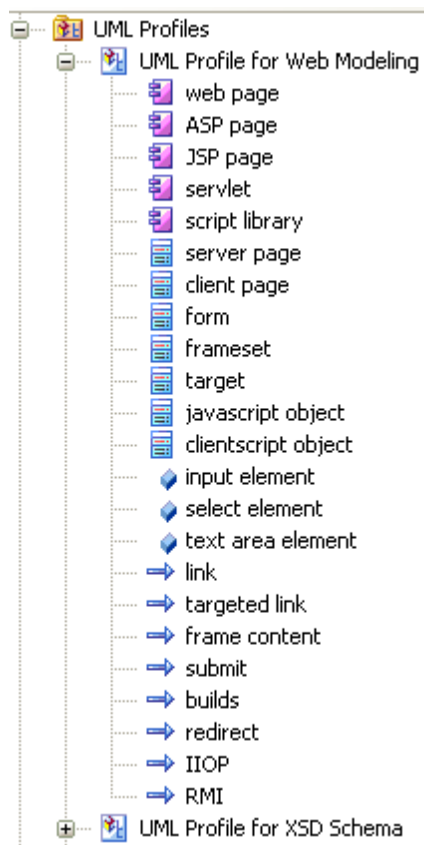
Once imported, you can drag and drop Profile elements onto the current diagram. Enterprise Architect attaches the stereotype, Tagged Values and default values, notes, and metafile if one is specified, to the new element. You can also drag and drop attributes and operations onto existing Classes and have them immediately added with the specified stereotype and values.

Profiles in the Resources Window:

The following describes how profiles are used in Enterprise Architect releases up to and including 6.5.

This method is still available in releases from 7.0 onwards; however, in these releases we recommend that you import and use profiles as a component of an MDG Technology. MDG Technologies allow a greater degree of customization of the user interface and enable profiles and their bundled resources to be shared more easily between projects and between users.

The Resources window contains a tree structure with entries for items such as MDG Technologies, Documents, Stylesheets, Matrix profiles and UML Profiles. The *UML Profiles* node initially contains no entries; to be able to use Profiles you must import them into Enterprise Architect from supplied XML files.



Items in the Profile represent stereotypes. These can be applied to UML elements in the following ways:

- Stereotypes that apply to elements such as *Classes* and *interfaces* can be dragged directly from the Resources window to the current diagram, automatically creating a stereotyped element; alternatively, they can be dragged onto existing elements, automatically applying them to the element
- Stereotypes that apply to *attributes* can be drag-and-dropped onto a host element (such as a Class); a stereotyped attribute is automatically added to the element's feature list
- Stereotypes that apply to *operations* are like those that apply to attributes; drag-and-drop onto a host element to add the stereotyped operation
- Stereotypes that apply to *connectors* such as *associations*, *generalizations*, *messages* and *dependencies* are added by selecting them in the Project Browser, then clicking on the start element in a diagram and dragging to the end element (in the same manner as adding normal connectors); a stereotyped connector is added
- Stereotypes that apply to *association ends* can be added by dragging the connector end element over the end of an Association in the diagram

Learn more

- [MDG Technology](#)^[1283]
- [MDG Technology SDK](#)^[1292]
- [Import a UML Profile](#)^[1279]

- [Add Profile Objects to a Diagram](#) ^[1280]
- [Tagged Values in Profiles](#) ^[1280]
- [Synchronize Tagged Values and Constraints](#) ^[1281]
- [Profiles webpage](#)

6.8.1 Import a UML Profile

To import a Profile you must have a suitable Profile XML file, such as the Profiles supplied on the Sparx Systems website. If the Profile includes references to any metafiles, they should be in the same directory as the Profile XML file.

How to

To import a Profile

Step	Action
1	Open the Resources window.
2	Right-click on the <i>UML Profiles</i> tree node and select the Import Profile context menu option. The Import UML Profile dialog displays.
3	Locate the XML Profile file to import using the (...) (Browse) button.
4	Set the required import option checkboxes for all stereotypes defined in the Profile; you can select: <ul style="list-style-type: none"> • Element Size - to import the element size attributes • Color and Appearance - to import the color (background, border and font) and appearance (border thickness) attributes • Alternate Image - to import the metafile image • Code Templates - to import the code templates if they exist • Overwrite Existing Templates - to overwrite any existing code templates defined in the current project
5	Click on the Import button. The Profile is added to the <i>UML Profiles</i> folder.

Notes

- If the Profile already exists, Enterprise Architect prompts you to overwrite the existing version and import the new one (or cancel); once the import is complete, the Profile is ready to use
- This topic describes importing a stand-alone Profile; you can also embed Profiles in an MDG

Technology and import the Technology file into the Enterprise Architect installation directory

Learn more

- [Profiles webpage](#)
- [Add Profile Objects and Features to a Diagram](#) ^[1280]
- [Creating MDG Technologies](#) ^[1324]
- [Deploying MDG Technologies](#) ^[1357]

6.8.2 Add Profile Objects to a Diagram

After you have imported a Profile into the Resources window, the profiled objects (elements and connectors) and features (attributes and operations) are available from the expanded *Profile* folder in the window and from the appropriate pages of the Toolbox (click on the **More tools** button at the top of the Toolbox).

Similarly, when you import an MDG Technology, it adds the appropriate pages of profiled elements and connectors to the Toolbox.

Action	Description
Add a Profile-based element to a diagram	Click on the element in the Toolbox page or the Resources window, and drag it onto the diagram.
Add a Profile-based connector to a diagram	Click on the connector in the Toolbox page or Resources window, then click on the source element in the diagram and drag it to the target. You can also drag the connector from the Resources window to the source, which automatically displays a list box of target elements; select a target from the list to create the connector to that target.
Add a profile-based attribute or operation to a diagram	Click on the attribute or operation in either the Toolbox page or the Resources window, and drag it onto the host element on the diagram. The system prompts you to enter a name for the feature.

Learn more

- [Toolbox](#) ^[698]
- [MDG Technology](#) ^[1283]

6.8.3 Tagged Values in Profiles

Stereotypes within a UML Profile can have one or more associated Tagged Values. When you create an element based on a UML Profile Stereotype by dragging from the Resources window to a diagram, any associated Tagged Values are added to the element as well. Tagged Values and Profiles are an excellent way to extend the use of Enterprise Architect and the power of UML modeling.

For example, in the UML Profile for XSD, there is an *XSDComplexType* stereotype, which has the following Tagged Value declaration:

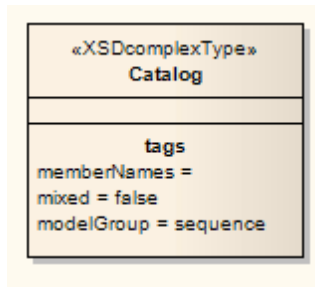
```
<TaggedValues>
```

```

<Tag name="mixed" description="Determines whether this element can
contain mixed element and character content. See the W3C XML Schema
recommendation" />
<Tag name="modelGroup" description="Overrides the package-level
default model group" values="all | sequence | choice"
default="sequence" />
<Tag name="memberNames" description="Overrides the package-level
default for naming complexType definitions" />
</TaggedValues>

```

When you create an element from the *XSDComplexType* stereotype (by dragging from the Profile Elements page of the Toolbox onto a diagram), the Tagged Values are added automatically.



Tagged Values that have default values are automatically set and displayed in the element *tags* section, if applicable. When you select the element, the Tagged Values window displays all the associated tags, including ones that have no value set. Also note that Tagged Values in the Profile that have a *Values* section (for example, *values="element | attribute | both" default="both"*) enable you to select the non-default values from a drop-down list. Where no *Value* list exists, the tag accepts free text.

Learn more

- [Predefined Structured Types](#) [1386]

6.8.4 Synchronize Tagged Values and Constraints

When you create an element, attribute, operation or link from a UML Profile item, you add the Tagged Values and constraints from the Profile. Over time you might modify the constraints or the notes and tags of the Tagged Values of a particular profiled item, so the items already created might be missing additional Tagged Value tags and notes, or constraints.

Similarly, you might have manually set the stereotype on a set of elements and now want them to receive the Tagged Values and constraints associated with that stereotype.

To make sure you have all the related Tagged Values and constraints, use the Synchronize Stereotype function. This operates on both:

- Profiles created in an MDG Technology File and not held in the Resources window - either an in-house customized Add-In, an imported external technology or an integrated technology provided with Enterprise Architect such as BPMN 1.1
- Profiles imported to the Resources window

When a Profile or MDG Technology file is deployed in Enterprise Architect, the Profile can be accessed through a set of Toolbox pages for that Profile or Technology. The profiled elements in these Toolbox pages automatically trigger an additional context menu option, **Synchronize Stereotype**.

How to

To synchronize elements using the Profile or Technology pages of the *Toolbox*

Step	Action
1	Ensure that the Toolbox displays pages containing the stereotyped Profile elements.
2	Right-click on the element profile in the Toolbox (for example, the BPMN 1.1 <i>Activity</i> element). The Toolbox context menu displays.
3	Click on the Synchronize Stereotype menu option. The Synch Profiled Elements dialog displays.
4	Click on the OK button to proceed. The Actions list is populated with the items that have been modified and the changes that were made.

Alternatively:

You can quickly synchronize the tags and constraints of a *single* element in a diagram by dragging the updated profile element from the Toolbox page onto the element in the diagram, then selecting the **Apply «stereotype»** context menu option.

Notes

- You can review any changes by displaying the element Properties dialog and by opening the Tagged Values window and clicking on an appropriate profiled element

6.9 MDG Technologies

The Model Driven Generation (MDG) Technologies enable you to access and use resources pertaining to a specific technology in Enterprise Architect. You have various options for bringing MDG Technologies into use with Enterprise Architect:

Source of Technology	See also
<p>Core technologies - Enterprise Architect consists of a:</p> <ul style="list-style-type: none"> • Basic UML 2 technology as an implementation of UML 2.3 structural and behavioral modeling, and • Core Extensions technology that applies profiles and stereotypes to provide extended modeling of aspects such as Requirements, User interface and Data Modeling 	
Sparx Systems already provides some added technologies in the Enterprise Architect Install directory, <i>MDGTechnologies</i> subfolder.	MDG Technologies ¹²⁸⁹
You can import technologies into the <i>APPDATA</i> folder (<i>%APPDATA%\Sparx Systems\EA\MDGTechnologies</i>).	Import MDG Technologies ¹²⁸⁸
You can transfer technologies into the <i>MDGTechnologies</i> subfolder; these technologies are available when you restart Enterprise Architect (on Vista/Windows 7 systems you might have to increase your access permissions to do this).	
You can access and activate MDG Technologies in remote system folders or web sites, from Enterprise Architect.	Access Remote Technologies ¹²⁸⁷
Technology Developers can create new MDG Technologies and deploy them to the project team either through the <i>MDGTechnologies</i> subfolder or from any other remote folder or website.	Create New Technologies ¹²⁹²

To see which technologies are available within Enterprise Architect, and activate the ones you require, use the MDG Technologies dialog.

Having made the MDG Technologies available to Enterprise Architect, you can manage their availability to users and you can work with them.

You also have the facility to turn off or disable the Enterprise Architect Basic UML 2 and Core Extensions Toolbox pages and facilities, so that you can apply the Enterprise Architect facilities and features exclusively to one or more selected MDG Technologies.

[Learn more](#)

- [Work with MDG Technologies](#) ^[1284]
- [Manage MDG Technologies](#) ^[1285]

6.9.1 Work with MDG Technologies

Any MDG Technology listed on the MDG Technologies dialog can be enabled, which makes their interface profiles and Toolbox pages available for your use.

Topic	Detail	See also
MDG Technology Toolbox Pages	<p>When you enable an MDG Technology, any Technology-specific diagram types are added to the New Diagram dialog lists, and the Technology's Toolbox pages are added to those available through the More tools menus in the Toolbox.</p> <p>If you set the MDG Technology to <i>Active</i>, its Toolbox pages override any parallel Enterprise Architect Toolbox pages; for example, the <i>ICONIX Class</i> pages would override the Enterprise Architect <i>Class</i> pages.</p> <p>You create Technology-specific diagrams and populate them with elements and connectors in the same way as for standard Enterprise Architect diagrams.</p>	Create New Diagrams ^[724]
The Resources Window	<p>A method of importing MDG Technologies into the Resources window is available (right-click on the <i>MDG Technologies</i> folder and select the Import Technology context menu option) but this method is Not Recommended.</p> <p>If you use this method, the MDG Technology is available only in the model it was imported into, and the Toolbox pages, Learning Center, Project Browser icons and model templates are not available.</p> <p>It is recommended that you instead create or download new technologies into the <code>%APPDATA%</code> folder or the Enterprise Architect installation directory, or use new technologies from remote file locations and web sites.</p> <p>However, you might previously have imported technologies into the Resources window, and these are still available until you specifically delete them (right-click on the technology and select the Delete Technology context menu option).</p>	Import MDG Technologies ^[1288] Access Remote MDG Technologies ^[1287]

The following information is provided to support implementations where technologies have been previously imported via the Resources window.

The Resources window (**View | More Project Tools | Project Resources**) displays a tree structure containing nodes such as imported MDG Technologies, Templates, Documents, Stylesheets, Matrix profiles and UML Profiles.

MDG Technologies can bundle the functionality provided by UML Profiles, UML Patterns, Code Templates and Model Types.

Profiles contained in MDG Technologies are applied to:

- Elements such as Classes and Interfaces, which are dragged directly from the Toolbox or the Resources window to the current diagram
- Attributes, which are dragged over a host element (such as a Class) to be automatically added to the element feature list
- Operations which, like Attributes, are dragged over a host element to add the operation
- Connectors such as Association, Generalization, and Dependency, which are added by selecting them in the Toolbox or Resources window, then clicking on the source element in a diagram and dragging to the target element (in the same way as adding normal connectors); the connector is added with the new stereotype and Tagged Value information
- Association Ends, which are added by dragging the connector end element over the end of an Association in the diagram

Patterns contained in MDG Technologies are used to:

- Enable reuse in a model
- Build in robustness

Code Templates are used to:

- Specify the transformation from UML elements into various parts of a given programming language


Model Types are used to:

- Define the data types for the model

6.9.2 Manage MDG Technologies

You use the MDG Technologies dialog to manage the MDG Technologies available and accessible to Enterprise Architect users.

Access **Settings | MDG Technologies**

Topic	Detail	See also
MDG Technologies Dialog	The MDG Technologies dialog lists the technologies held in the <i>APPDATA</i> folder (marked by the text <i>Location: Technology.xml</i>) and Enterprise Architect Install directory, in alphabetical order.	Import MDG Technologies 
Enable and Disable MDG Technologies	All MDG Technologies listed can be made available (enabled) or removed from use (disabled); to enable or disable a Technology, click on its Enabled checkbox. When an MDG Technology is enabled, three things happen:	

Topic	Detail	See also
	<ul style="list-style-type: none"> • The MDG Technology is added to the list of available options in the profile field of the Default Tools toolbar, so that you can apply the interface profiles of the MDG Technology • At least one set of Toolbox pages for the MDG Technology is automatically added to the Toolbox; you can access the added Toolbox pages through the More Tools option • Any MDG Technology-specific diagram templates are added to the New Diagram dialog for selection; when selected, these display the diagram-specific Toolbox pages <p>You can quickly enable or disable all the listed MDG Technologies by clicking on the All or None buttons.</p> <p>However, if you click on the None button, you should scroll to the top of the list and select the Basic UML 2 Technology and Core Extensions checkboxes to re-enable the UML and Extended Toolbox pages and diagram types.</p>	
Set as Default	<p>You can make an MDG Technology the default interface to Enterprise Architect. Depending on the MDG Technology selected, this can change the way Enterprise Architect windows are displayed and override the Toolbox pages with pages specific to that Technology.</p> <p>To set an MDG Technology as the default interface, click on it in the Technology panel and click on the Set Active button.</p> <p>This displays an asterisk against the MDG Technology name in the Technology panel, and selects the MDG Technology in the profile field of the Default Tools toolbar. If the MDG Technology has not been enabled, this also enables it.</p>	
Basic UML 2 Technology	<p>You can also enable one or more of the MDG Technologies (and likely make one of them the default) and then <i>deselect</i> the Basic UML 2 Technology checkbox (and, if necessary, the Core Extensions checkbox), to work exclusively in the selected technologies. The UML Toolbox pages, diagram types and quicklinks are excluded from the Toolbox, More tools option, diagrams and New Diagram dialog in the user interface.</p>	
MDG Technologies Outside	<p>The MDG Technologies dialog lists technologies that have been loaded into the <i>APPDATA</i> folder and Enterprise Architect install directory.</p>	<p>Access Remote MDG Technologies^[1287]</p>

Topic	Detail	See also
Enterprise Architect	You can also add MDG Technologies in folders and websites remote from Enterprise Architect. To do this, click on the Advanced button.	

6.9.3 Access Remote MDG Technologies

You can access MDG Technologies in folders and websites remote from Enterprise Architect.

If you have not already identified the location of the MDG Technology, you must first do this.

Later, if you have no further use for the MDG Technology, you can remove it from the list of identified MDG Technologies.

If you add or remove remote MDG Technologies, you must *restart* Enterprise Architect to show them on, or remove them from, the list on the MDG Technologies dialog, and to use them.

Access **Settings | MDG Technologies**

How to

To specify the location of a remote MDG technology

Step	Action	See also
1	Select the MDG Technologies menu option. The MDG Technologies dialog displays.	Manage MDG Technologies ^[1285]
2	Click on the Advanced button. The MDG Technologies - Advanced dialog displays.	
3	Click on the Add button. A short context menu displays, offering the options: <ul style="list-style-type: none"> • Add Path • Add URL 	
4	To specify an MDG Technology <i>in a directory folder</i> , select the Add Path option. The Browse for Folder dialog displays. Browse for the MDG Technology folder, click on it, and click on the OK button; go to step 6.	
5	To specify an MDG Technology <i>on a web site</i> , select the Add URL option. The Input dialog displays.	

Step	Action	See also
	In the Enter Value field, type or copy-and-paste the MDG Technology URL and click on the OK button.	
6	The folder path or URL for the MDG Technology displays in the Path panel.	

To remove an MDG Technology listed in the MDG Technologies - Advanced dialog, click on the folder path or URL and click on the **Remove** button; the path or URL is deleted.

6.9.4 Import MDG Technologies to APPDATA

The preferred method of making an MDG Technology available to your project team members is to import the technology into the %APPDATA%\Sparx Systems\EA\MDGTechnologies folder. To import an MDG Technology you must have a suitable MDG Technology XML file. If the MDG Technology includes references to any metafiles, they should be in the same directory as the MDG Technology XML file.

On startup, Enterprise Architect scans the *APPDATA* folder (as well as the Enterprise Architect Install directory *MDGTechnologies* subfolder) for technology files, to make them available through the MDG Technologies dialog. Technologies imported to the *APPDATA* folder are indicated by the text *Location: Technology.xml*.

Access [Tools | MDG Technology Import](#)

How to

To import a technology

Step	Action
1	Select the MDG Technology Import menu option. The Copy Technology to Application Data dialog displays.
2	In the Filename field, type the path and filename of the MDG Technology file to import, or browse for it using the (...) button. When you enter the filename, the MDG Technology name and version display in the Technology and Version fields.
3	Click on the OK button. <ul style="list-style-type: none"> • If the <i>APPDATA</i> folder does not yet exist, Enterprise Architect creates it • If the MDG Technology already exists, Enterprise Architect displays a prompt to overwrite the existing version and import the new one <p>Once the import is complete, you must restart Enterprise Architect; the MDG Technology is then listed in the MDG Technologies dialog.</p>

Step	Action

Notes

- This feature is not available in the Desktop edition of Enterprise Architect

Learn more

- [Manage MDG Technologies](#) ^[1285]

6.9.5 Extensions - MDG Technologies

Enterprise Architect is the core for a range of Model Driven Generation (MDG) extensions to its modeling capabilities, using more specialized, niche frameworks and profiles.

Topic	See also
A number of technologies are already integrated with the Enterprise Architect installer.	ArchiMate ^[1621] BPEL ^[1570] BPMN ^[1545] Data Flow Diagrams ^[1506] Eriksson-Penker Extensions ^[1623] ICONIX ^[1925] Mind Mapping ^[1503] SoaML ^[2088] SOMF ^[2092] Strategic Modeling Systems Modeling Language (SysML) ^[1990]
Enterprise Architect provides support for: <ul style="list-style-type: none"> • Downloading MDG Technologies from external system files or websites, or • Creating your own easily with the Enterprise Architect MDG Technology Wizard 	Access Remote MDG Technologies ^[1287] MDG Technology SDK ^[1292]
Sparx Systems also market a number of MDG products, as follows: <ul style="list-style-type: none"> • MDG Technology For: <ul style="list-style-type: none"> • Zachman Framework • The Open Group Architecture Framework (TOGAF) • Unified Profile for DoDAF and MODAF (UPDM) 	

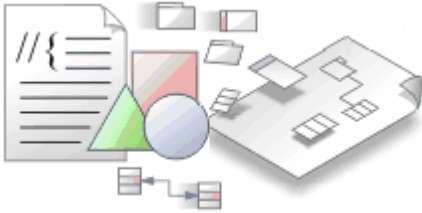
Topic	See also
<ul style="list-style-type: none"> • Data Distribution Service (DDS) • Python (for Enterprise Architect versions 4.5 to 5.0, integrated in later versions) (* free product! *) • CORBA (* free product! *) • Java Beans (* free product! *) • Testing (* free product! *) • MDG Integration For: <ul style="list-style-type: none"> • Eclipse 3.3 • Visual Studio 2005 and 2008 • Siemens PLM Teamcenter Systems Engineering (TcSE) • MDG Link For <ul style="list-style-type: none"> • Eclipse • Visual Studio.NET • Microsoft Visio (* free product! *) • Telelogic DOORS <p>Over time, this list is being extended to include further products.</p>	
<p>Sparx Systems provide different editions of Enterprise Architect tailored for systems engineering and business engineering, or both together.</p> <p>These editions incorporate several of the above MDG Technologies and other Add-Ins.</p>	Enterprise Architect Editions ^[20]
<p>Product Information: For the latest list of available Add-Ins and an introduction to each product, including details of pricing, purchasing and download options, see the Sparx Systems website.</p> <p>When you purchase one of the Add-Ins, you receive one or more license keys and instructions on obtaining, installing and registering the product.</p> <p>The information page for most products provides a link to download the product User Guide in .pdf format.</p> <p>The product User Guide can also be displayed as a .chm file online within the product itself; to access this online help in Enterprise Architect, select the Extensions <productname> Help menu option.</p>	Products Page on the Sparx Systems website

Learn more

- [Access Remote MDG Technologies](#) ^[1287]

- [Creating MDG Technologies](#) 

6.10 Define a Modeling Language



Enterprise Architect enables you to create models using UML. However, it also enables you to go much further, extending the scope both of your modeling and of the UML components you use, as outlined below.

Topic	Description	See also
UML Stereotypes	<p>Stereotypes are an inbuilt mechanism for logically extending or altering the meaning, display and syntax of a model element.</p> <p>Different model elements have different standard stereotypes associated with them. You can also define your own stereotypes.</p>	UML Stereotypes ¹²⁶⁴
UML Profiles	<p>UML Profiles are a means of extending UML, which enables you to build models in particular domains.</p> <p>A Profile is a collection of additional stereotypes and Tagged Values applied to elements, attributes, methods and connectors, which together describe some particular modeling problem and facilitate modeling constructs in that domain.</p>	UML Profiles ¹²⁷⁷
UML Patterns	<p>Patterns are groups of collaborating Objects/Classes that can be abstracted from a general set of modeling scenarios (that is, parameterized collaborations).</p> <p>They generally describe how to solve an abstract problem, and are an excellent means of achieving re-use and building in robustness.</p>	Design Patterns ¹²⁷¹
MDG Technologies	<p>The Model Driven Generation (MDG) Technologies enable you to access and use the resources of a specific technology within Enterprise Architect.</p> <p>Interfaces to some technologies, such as BPMN and ICONIX, are integrated with Enterprise Architect, whilst interfaces to others such as Eclipse and Visual Studio can be added separately. You can also link to technologies that you have created yourself.</p>	MDG Technologies ¹²⁸³

6.10.1 MDG Technology SDK

Introduction

In describing aspects of developing technologies to use in conjunction with Enterprise Architect, it is

expected that you are familiar with the concepts introduced in the main body of the *Enterprise Architect User Guide*. Wherever appropriate, links to these concepts are provided in the topics.

Contents:

- [Developing Profiles](#)^[1293] (incorporating [Custom Stereotypes](#)^[1293])
- [MDG Technologies](#)^[1323]
- [Shape Scripts](#)^[1358]
- [Tagged Value Types](#)^[1385]
- [Code Template Framework](#)^[1394]

6.10.1.1 Developing Profiles

Introduction

Profiles provide a means of extending the UML, which enables you to build models in particular domains. They are based on additional stereotypes and Tagged Values that are applied to UML elements, connectors and their components. A Profile is a collection of such extensions that together describe some particular modeling problem and facilitate modeling constructs in that domain. UML Profiles for Enterprise Architect are specified in XML files, with a specific format. These XML files can be imported into Enterprise Architect as a component of an MDG Technology.

The imported Profile also automatically generates a page of elements and relationships in the Toolbox.

Items in the Profile represent stereotypes. UML supports a large number of stereotypes, which are an inbuilt mechanism for logically extending or altering the meaning, display, appearance and syntax of a model element. Different model elements have different stereotypes associated with them.

Learn more

- [MDG Technologies](#)^[1283]
- [Custom Stereotypes](#)^[1293]
- [Create Profiles](#)^[1295]
- [Quick Linker](#)^[1317]
- [Customize Toolbox Profiles](#)^[1337]
- [Create Diagram Profiles](#)^[1346]
- [Create Learning Center Profiles](#)^[1350]
- [Tagged Value Types](#)^[1385]
- [UML Profiles](#)^[1277] (for more information on the use of Profiles in Enterprise Architect)

6.10.1.1.1 Custom Stereotypes

UML supports a large number of *stereotypes*, which are an inbuilt mechanism for logically extending or altering the meaning, physical appearance and syntax of a model element. Different model elements have different stereotypes associated with them.

In Enterprise Architect you can create new stereotypes with their own custom appearance. The stereotypes can be altered to make use of metafiles (image files) and customized colors, or you can make use of an Enterprise Architect Shape Script to make new element shapes to determine the shape and dimensions of the element.

Access [Settings | UML Types > Stereotypes](#)

How to

To add your own custom stereotypes

Step	Action
1	On the Stereotypes tab, type or select a Stereotype name.
2	Select a Base Class from the drop-down list.
3	Associate a Metafile with this stereotype - click on the Metafile radio button and the Assign button, and locate the required .emf or .wmf file.
4	Enter optional Notes and select Default Colors for this stereotype.
5	Click on the Save button to save the stereotype.

The table below describes the functions of the Stereotypes tab.

Field	Usage	See also
Stereotype	Specify the name of the stereotype.	
Group name	Enable grouping of stereotype features by a plural name, for attributes and operations, which is shown on diagrams in the attribute and operations compartments.	
Base Class	Enable the stereotyped element to inherit the base characteristics from a pre-existing element type.	
Notes	Type any notes concerning the stereotype (not the elements to which the stereotype is to be applied).	
Override Appearance		
None	Switch to the default element appearance.	

Field	Usage	See also
Metafile	Enable an image file to be used for the appearance of the stereotype.	
Shape Script	Specify custom shapes for the stereotype using the Enterprise Architect Shape Scripting language.	Shape Scripts <small>[1358]</small>
Assign	Add the associated metafile or Shape Script from the stereotyped element.	
Remove	Remove the associated metafile or Shape Script from the stereotyped element.	
Default Colors		
Fill	Set the default background color of the element.	
Border	Control the border color.	
Font	Control the color of the stereotype font.	
Reset	Reset the appearance of the element to the default element appearance.	

Notes

- You can transport custom stereotype definitions between models, using the Export Reference Data and Import Reference Data options

Learn more

- [UML Stereotypes](#) [1264]
- [Export Reference Data](#) [306]
- [Import Reference Data](#) [308]

6.10.1.1.2 Create Profiles

This topic describes how to create profiles and profile items. These creation tasks include creating the profile stereotypes, defining the metaclasses they apply to, and defining Tagged Values and constraints. The topic also describes how to export a profile for use in modeling.

How to

To create a Profile

Step	Description	See also
1	Create a Profile package.	Create Profile ^[1296]
2	Add stereotypes and metaclasses to the profile.	Add Stereotypes and Metaclasses ^[1297]
3	Define Tagged Values for the stereotypes.	Define Stereotype Tags ^[1299]
4	Define constraints for the stereotypes.	Define Stereotype Constraints ^[1302]
5	Add enumeration elements to define values for the Tagged Values for the stereotypes.	Add Enumeration Elements ^[1303]
6	Add shape scripts for the stereotypes.	Add Shape Scripts ^[1304]
7	Set the default appearance for the stereotypes.	Set Default Appearance ^[1306]
8	Export the profile.	Export Profile ^[1308]

6.10.1.1.2.1 Create a Profile Package

In Enterprise Architect, you must create a Profile in a Package that has the stereotype «*profile*».

How to

To create a Profile Package

Step	Description	See
1	Open or create a Package diagram.	
2	Open the Profile page of the Toolbox (More tools Profile).	Profile Group ^[714]
3	Drag the <i>Profile</i> item onto the Class diagram. The New Model Package dialog displays.	
4	In the Package Name field, type a name for the Profile.	

Step	Description	See
5	Select the Automatically add new diagram checkbox.	
6	Click on the OK button. The New Diagram dialog displays.	New Diagram Dialog [724]
7	Provide the required diagram name, and select the diagram group UML Structural and diagram type Class .	
8	Click on the OK button. Enterprise Architect creates a package with the stereotype « <i>profile</i> » and with a child Class diagram.	
9	In the Project Browser, double-click on the Profile Package on the diagram to open the child diagram. You now use this child diagram to add stereotypes to the Profile.	

Learn more

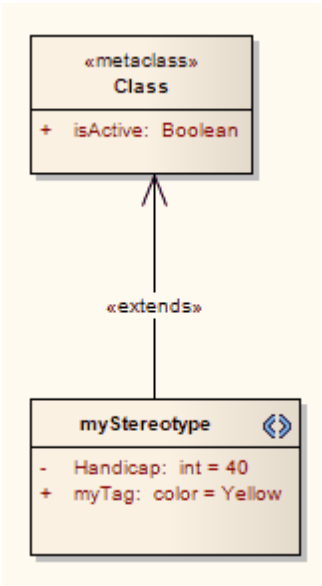
- [Add Stereotypes and Metaclasses](#) [1297]

6.10.1.1.2.2 Add Stereotypes and Metaclasses

How to

To add metaclasses and stereotypes to a Profile, for as many stereotypes and metaclasses as you require

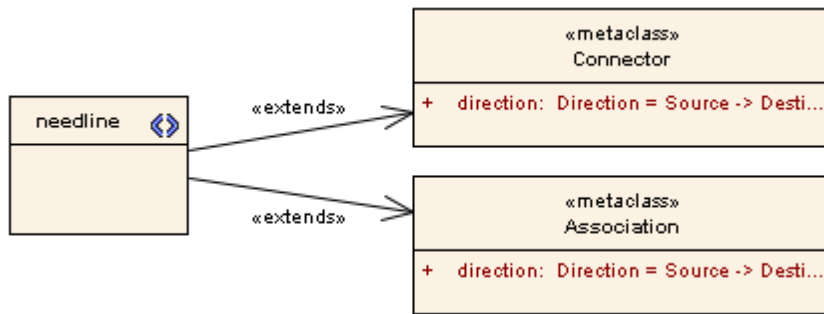
Step	Description	See also
1	Open the child diagram of the Profile Package.	
2	Drag the <i>Metaclass</i> element from the Profile page of the Toolbox onto the diagram. The Create New Metaclass dialog displays, in which you can tick multiple metaclasses for dropping onto the diagram.	
3	Scroll down the Element list and select the checkbox for Class .	
4	Click on the OK button. The Class Properties dialog displays.	

Step	Description	See also
5	In the Name field type a name for the element. Click on the OK button.	
6	Drag a <i>Stereotype</i> element from the Toolbox onto the diagram. If the Properties dialog does not display, double-click on the element on the diagram.	
7	In the Name field, type a name for the stereotype.	
8	Click on the OK button.	
9	Click on the <i>Extension</i> relationship in the Toolbox and drag the connection from the stereotype element to the metaclass element.	
10	Your diagram should now resemble the one below:  <pre> classDiagram class Class { <<metaclass>> + isActive: Boolean } class myStereotype { - Handicap: int = 40 + myTag: color = Yellow } Class < -- myStereotype </pre>	

Notes

- If you want to have a stereotype extending more than one metaclass, do not create two stereotype Classes with the same name

You cannot have two stereotypes with the same name in the same profile, one is discarded when you save the profile; therefore, create one stereotype Class with an Extension connector to each of several Metaclass elements, as shown below.



You can now add stereotype Tags, Constraints, Enumerations, and/or Shape Scripts to your Profile, and define the default appearance of the elements or connectors as required

Learn more

- [Define Stereotype Tags](#) ^[1299]
- [Define Stereotype Constraints](#) ^[1302]
- [Adding Enumeration Elements](#) ^[1303]
- [Adding Shape Scripts to UML Profiles](#) ^[1304]
- [Setting Default Appearance in UML Profiles](#) ^[1306]

6.10.1.1.2.3 Define Stereotype Tagged Values

Stereotypes within a UML Profile can have one or more associated Tagged Values. When creating a UML Profile, you define these Tagged Values as attributes of the stereotyped Class.

How to

To define Tagged Values for a stereotype

Step	Description	See also
1	Open the Attributes dialog for the stereotyped element.	
2	Click on the New button to create a new attribute.	
3	In the Name field, type the name of the stereotype tag.	
4	In the Type field, click on the drop-down arrow and select the attribute type.	
5	In the Initial field, type the initial value of the tag.	Add Enumeration Elements ^[1303] (for creating enumerated types for Tagged Values)

Step	Description	See also
6	In the Notes field, type a description of the tag.	
7	Click on the Save button and Close button.	

Learn more

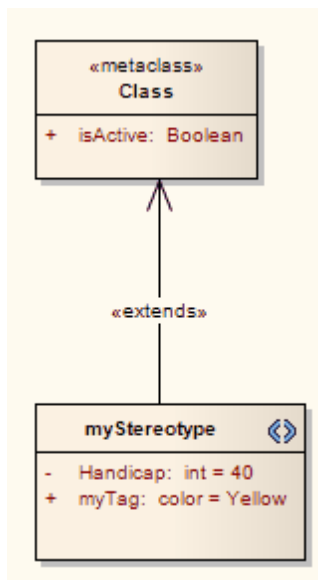
- [Define Stereotype Tags with Predefined Tag Types](#)^[1300]
- [Define Stereotype Tags with Supported Attributes](#)^[1301]
- [Use the Tagged Value Connector](#)^[1301]

Define Predefined Tag Types:

To define a stereotype tag with a predefined Tagged Value Type, you must first create the predefined Tagged Value Type. For full instructions on how to do this, see the *Create Structured Tagged Values* topic.

Assign Predefined Tag Types to Stereotypes:

To assign a predefined tag type to a stereotype, just create an attribute with the same name. For example, to make the Tagged Value *Handicap* appear in a stereotype, create an attribute named *Handicap*. You can set the default value for the Tagged Value by giving the attribute an *Initial* value.



Learn more

- [Create Structured Tagged Values](#)^[1389]

Supported stereotype attribute tags are special tags that set the default behavior of stereotyped elements, such as the initial size of the element and the default location of any image files associated with the stereotype.

How to

To define tags for a stereotype with supported attributes

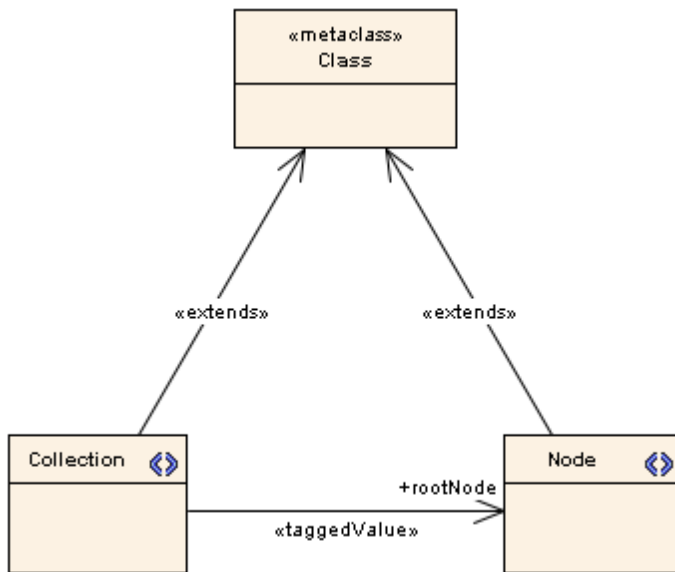
Step	Description	See also
1	Open the Attributes dialog for the stereotyped element.	
2	In the Name field, type the name of the stereotype tag.	
3	In the Initial field, type the initial value of the tag.	
4	For supported attributes you set only the Name (which must match the attributes listed in the supported attributes section) and the Initial value; do not set the other values. Click on the Save button and Close button.	

Learn more

- [Supported Attributes](#) 

In a Profile, you can use the *Tagged Value* connector to define a Tagged Value that has as its value the name of an element containing the stereotype pointed to. You select the Tagged Value connector from the Profile pages of the Toolbox.

The following diagram demonstrates how you might use the connector. It shows a (*saved* and *imported*) profile that defines two stereotypes: «*Collection*» and «*Node*». The «*Collection*» stereotype has a Tagged Value connector with the target role named *rootNode*, pointing to the «*Node*» stereotype.



In the Tagged Values window for the connector, against *rootNode*, you click on the selection button ((...)). This displays the Select <Item> dialog, through which you locate the elements in the current model with the «Node» stereotype. You can then select one of these elements as the value of the tag.

Learn more

- [Select <Item> Dialog](#)^[880]

6.10.1.1.2.4 Define Stereotype Constraints

Defining constraints for stereotypes uses the same procedure as defining constraints for any Class.

How to

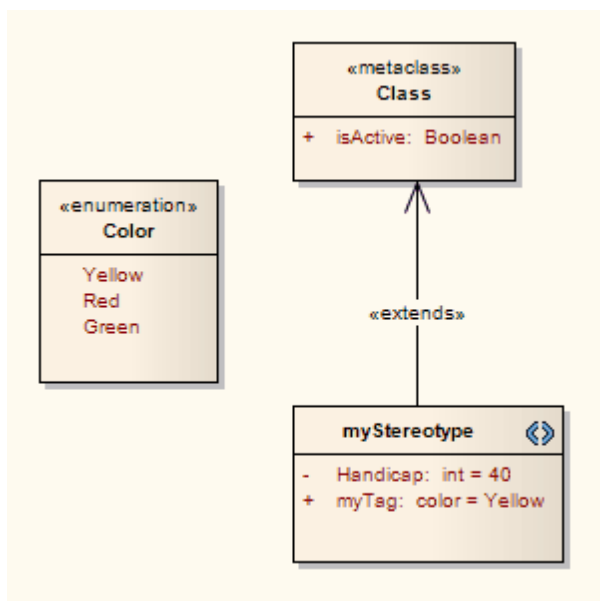
To define constraints for a stereotype

Step	Description	See also
1	Open the Class Properties dialog of the stereotype element in a diagram.	
2	Click on the Constraints tab and click on the New button to create a new constraint.	
3	In the Constraint field, type the value of the constraint.	
4	In the Type field, click on the drop-down arrow and select the appropriate type.	
5	In the Status field, click on the drop-down arrow and select the appropriate status.	

Step	Description	See also
6	In the Notes field, type any additional information required.	
7	Click on the Save button, and on the OK button to close the dialog.	

6.10.1.1.2.5 Add Enumeration Elements

Enumerations can be used to restrict the values available to stereotype tags.



How to

To add an Enumeration element

Step	Description	See also
1	Open your Profile Package child Class diagram.	
2	In the Toolbox, select More tools Profile . The contents of the Profile page of the Toolbox display.	
3	Drag an <i>Enumeration</i> item from the toolbox onto the diagram. If the Properties dialog does not display, double-click on the element on the diagram.	
4	In the Name field, type the name of the new Enumeration.	


Step	Description	See also
5	Click on the Details tab and on the Attributes button. The Attributes Properties dialog displays.	
6	In the Name field, type the name of the Enumeration attribute.	
7	In the Type field, click on the drop-down arrow and select the appropriate type.	
8	In the Initial field, type the initial value of the attribute.	
9	Click on the Save button, and repeat steps 6 to 9 for additional attributes.	
10	When you are finished, click on the Close button.	
11	Right-click on the <i>Stereotype</i> element and select the Attributes context menu option. The Attribute Properties dialog displays for the <i>stereotype</i> .	
12	In the Name field type a name for the attribute.	
13	In the Type field click on the (...) button and select the name of the enumeration element from the Select <Item> dialog.	
14	In the Initial field type the name of the required enumeration attribute as the default value.	
15	Click on the Save and Close buttons. You have now generated a drop-down list for setting the value of the tag in the Tagged Values window.	

6.10.1.1.2.6 Add Shape Scripts

How to

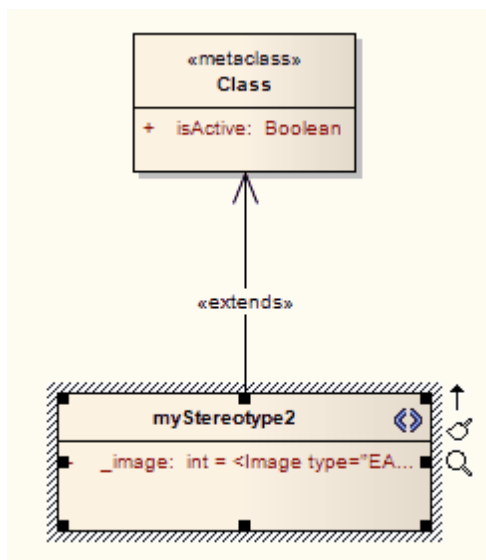
To add a Shape Script to a stereotype in a UML Profile (such as an MDG Technology Stereotype Profile)

Step	Description	See also
1	On the Profile Package child diagram, select a <i>Stereotype</i> element.	

Step	Description	See also
2	Right-click on the element and select the Attributes context menu option.	
3	In the Attributes Properties dialog, in the Name field, type _image .	
4	Click on the (...) button next to the Initial field. The Shape Editor dialog displays.	Shape Editor 
5	Enter the Shape Script in the Shape Editor dialog. Click on the OK and Close buttons.	

In some cases, particularly MDG Technology Stereotype Profiles, your Shape Script might include externally-defined images. In this case the Shape Script would include the image method, specifying the image file name prefixed with the technology name.

The Stereotype element now resembles the example below:



Notes

- If you are creating a Shape Script for an Association Class, be aware that the Shape Script is applied to both the Class part and the Association part; therefore, you might have to include logic in the shape main that tests the type of the element so that you can give separate drawing instructions for Class and for Association

Such logic is not necessary in the:

- shape source or shape target, which are ignored by Classes, or the
- decoration shapes which are ignored by Associations

Learn more

- [MDG Technology Stereotype Profiles](#) ^[1316]
- [Shape Scripts](#) ^[1358]
- [Drawing Methods](#) ^[1366]

6.10.1.1.2.7 Set Default Appearance

You can define the appearance of stereotyped elements and connectors as you create or edit the stereotypes, using the Override Appearance and Default Colors panels of the UML Types dialog.

However, an easier way is to review your completed profile diagram and set the default appearance of the elements and connectors in place.

Simply click on the required element or connector and press (**F4**), then define the background, font and border colors and border thickness as appropriate, on the Default Appearance dialog.

When you save the profile containing the stereotyped elements and connectors, make sure that you select the **Color and Appearance** checkbox on the Save UML Profile dialog.

Learn more

- [Creating Custom Stereotypes](#) ^[1293]
- [Configure Default Appearance](#) ^[817]
- [Saving Profiles](#) ^[1306]

6.10.1.1.2.8 Export a Profile

Once you have created a Profile and defined the elements and metaclasses, you can save (export) the Profile to disk for future models.

How to

To save a Profile

Step	Description	See also
1	<p>If your profile is:</p> <ul style="list-style-type: none"> • A single profile spread over multiple diagrams within the same Profile package, find the Profile package in the Project Browser window, right-click on it and select the Save Package as UML Profile context menu option • One of multiple profiles within the same Profile package, right-click anywhere in the background of the Profile diagram and select the Save as Profile context menu option 	<p>Save Profile Options ^[1307]</p>

Step	Description	See also
	<ul style="list-style-type: none"> A single diagram within the Profile package, choose either the Save Package as UML Profile context menu option or the Save as Profile context menu option <p>The Save UML Profile dialog displays.</p>	
2	<p>Click on the (...) (Browse) button, and select the export destination for the XML Profile file.</p> <p>If necessary, edit the profile filename, but do not delete the .xml extension.</p>	
3	<p>In the Profile Type field, use the default value EA (UML)2.X (or, if necessary, click on the drop-down arrow and select this value).</p>	
4	<p>Set the required export options for all stereotypes defined in the profile:</p> <ul style="list-style-type: none"> Element Size - select the checkbox to export the element size attributes Color and Appearance - select the checkbox to export the color (background, border and font) and appearance (border thickness) attributes Alternate Image - select the checkbox to export the metafile images Code Templates - select the checkbox to export the code templates, if they exist 	Color and Appearance <small>1306</small>
5	<p>Click on the Save button to save the profile to disk.</p>	

Learn more

- For information on importing and using the profile in modeling, see the [UML Profiles](#)1277 topic

When you save a UML Profile, you can save it either from the package or from the diagram, depending on whether the Profile is:

- A single profile spread over multiple diagrams within the same Profile package (find the Profile *package* in the Project Browser, right-click on it and select the **Save Package as UML Profile** context menu option), which is typically the case for a stereotypes profile
- One of multiple profiles within the same Profile package (right-click anywhere in the background of the Profile *diagram* and select the **Save as Profile** context menu option); for example, when creating multiple toolbox profiles
- A single diagram within the Profile Package (choose *either* the **Save Package as UML Profile** context menu option *or* the **Save as Profile** context menu option)

The two context menu options produce slightly different results. You should take these into consideration, especially in the third instance where you could choose either option.

Save From Diagram	Save From Package	Notes
The profile takes the diagram name.	The profile takes the package name.	Package and diagram names are not necessarily the same, although you can save a lot of confusion if you make them the same or very similar. For example: package <i>GL</i> with diagrams <i>GL1</i> , <i>GL2</i> , <i>GL3</i> .
The profile takes the diagram's notes.	The profile takes the package's notes.	
You can take the default size and appearance (including alternate image) from the diagram object.	You cannot take the default size and appearance from the diagram object. You can use the <i>_sizeX</i> , <i>_sizeY</i> and <i>_image</i> properties, but there is no equivalent for default colors.	
This option can be much faster.	This option can be much slower	The difference arises because diagram objects are kept in memory and Project Browser elements are not. This is only likely to be an issue if the profile is a large one and you are using a slow network connection to a remote repository.

6.10.1.1.2.9 Supported Attributes

Supported Stereotype Attributes:

The following attributes can be applied to stereotypes in UML Profiles:

Attribute	Meaning	See also
icon	Contains the path to a bitmap file to be used as the Project Browser icon for all elements other than <i>Package</i> , with the given stereotype. The bitmap must be 16x16 pixels. For a transparent background, use light grey - RGB(192,192,192). For this attribute to work correctly, the <i>_metatype</i> attribute must also be used (see below).	

Attribute	Meaning	See also
_image	Shape script definition.	
_instanceMode	Used for defining behavior on creating an instance.	Defining Behavior on Creating an Instance <small>[1313]</small>
_instanceOwner		
_instanceType		
_metatype	Used for defining stereotypes as metatypes.	Define a Stereotype as a Metatype <small>[1312]</small>
_sizeY	Initial height of the element, in pixels at 100% zoom.	
_sizeX	Initial width of the element, in pixels at 100% zoom.	
_strictness	Used for restricting application of multiple stereotypes.	Restricting Application of Multiple Stereotypes <small>[1313]</small>

Supported Metaclass Attributes:

The following attributes can be applied to metaclass Classes in UML Profiles, and refer to the stereotypes that extend them:

Attribute	Meaning	See also
_AttInh	If set to 1 , switches on the <i>Inherited Features: Show Attributes</i> setting.	
_AttPkg	If set to 1 , switches on the <i>Attribute Visibility: Package</i> setting.	
_AttPri	If set to 1 , switches on the <i>Attribute Visibility: Private</i> setting.	
_AttPro	If set to 1 , switches on the <i>Attribute Visibility: Protected</i> setting.	
_AttPub	If set to 1 , switches on the <i>Attribute Visibility: Public</i> setting.	

Attribute	Meaning	See also
_ConInh	If set to 1 , switches on the <i>Show Element Compartments: Inherited Constraints</i> setting.	
_Constraint	If set to 1 , switches on the <i>Show Element Compartments: Constraints</i> setting.	
_defaultDiagramType	Used for defining child diagram types.	Define Child Diagram Types ^[1315]
_HideStype	If set to a comma-separated list of stereotypes, sets the <i>Hide Stereotyped Features</i> filter.	
_isVertical	If set to True for an <i>ActivityPartition</i> stereotype, creates a vertical <i>ActivityPartition</i> by default.	
_lineStyle	Sets the line style of a connector; the value of the attribute can be one of: <ul style="list-style-type: none"> • direct • auto • custom • bezier • treeH (horizontal) • treeV (vertical) • treeLH (lateral horizontal) • treeLV (lateral vertical) • orthogonalIS (orthogonal, square corners) • orthogonalIR (orthogonal, rounded corners) 	
_makeComposite	Used for creating composite elements.	Create Composite Elements ^[1314]
_OpInh	If set to 1 , switches on the <i>Inherited Features: Show Operations</i> setting.	
_OpPkg	If set to 1 , switches on the <i>Operation Visibility: Package</i> setting.	
_OpPri	If set to 1 , switches on the <i>Operation Visibility: Private</i> setting.	
_OpPro	If set to 1 , switches on the <i>Operation Visibility: Protected</i> setting.	
_OpPub	If set to 1 , switches on the <i>Operation Visibility: Public</i> setting.	

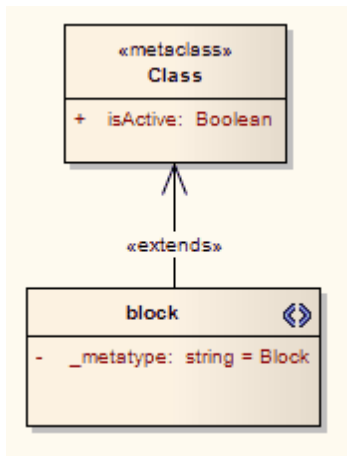
Attribute	Meaning	See also
_PType	If set to 1 , switches on the <i>Show element type (Port or Part only)</i> setting.	
_ResInh	If set to 1 , switches on the <i>Show Element Compartments: Inherited Responsibilities</i> setting.	
_Responsibility	If set to 1 , switches on the <i>Show Element Compartments: Responsibilities</i> setting.	
_Runstate	If set to 1 , switches on the <i>Hide Object Runstate in current diagram</i> setting.	
_SourceAggregation	Used to set the aggregation type at the end of a connector; do not set <i>both</i> _SourceAggregation and _TargetAggregation . Set to 1 for shared, 2 for composite.	
_SourceMultiplicity	Used to set the multiplicity of the source element, such as 1..* or 0..1 .	
_SourceNavigability	If the connector is non-navigable, set this attribute to Non-Navigable ; for other values, set the Direction attribute.	Connector Class ^[2508]
_SubtypeProperty	Used to specify the fully qualified name of a Tagged Value that acts as a subtype for the element, and that must be set when created. This Tagged Value must be created on the stereotype using an Association to an Enumeration element.	
_Tag	If set to 1 , switches on the <i>Show Element Compartments: Tags</i> setting.	
_tagGroupings	Maps the Tagged Values into the _TagGroups (below) displayed in the Tagged Values window, in the form: <i>tagName1=groupName1;tagName2=groupName2;</i>	
_tagGroups	A comma-separated list of required groups in the order in which they are to be displayed in the Tagged Values window <i>groupName1,groupName2</i>	
_tagGroupStates	Maps _tagGroups displayed in the Tagged Values window to the state of open or closed , in the form: <i>groupName1=open;groupName2=closed;</i>	

Attribute	Meaning	See also
_TagInh	If set to 1 , switches on the <i>Show Element Compartments: Inherited Tags</i> setting.	
_TargetAggregation	Used to set the aggregation type at the end of a connector; do not set <i>both</i> _SourceAggregation and _TargetAggregation . Set to 1 for shared, 2 for composite.	
_TargetMultiplicity	Used to set the multiplicity of the target element, such as 1..* or 0..1 .	
_TargetNavigability	If the connector is non-navigable, set this attribute to Non-Navigable ; for other values, set the Direction attribute.	Connector Class <small>[2508]</small>

The *_metatype* attribute is applied to a stereotype element. This is used where users want to hide the identity of an element as a stereotyped UML element. It is also a method of getting custom types to appear in contexts where only Enterprise Architect's inbuilt types would normally appear; for example in the lists of element types in the Relationship Matrix.

The **Strict UML Syntax check** is not performed for stereotyped elements that have had the *_metatype* attribute set.

In the following example from SysML, *block* is defined as a stereotype that extends a UML Class.



However, a SysML user is not interested in UML Classes, only in SysML Blocks. An element created from a stereotype defined this way, while behaving like a stereotyped Class in most contexts:

- Shows *Block Properties* rather than *Class Properties* as the title of its Properties dialog
- Is auto-numbered as *Block1* not *Class1* on creation, and
- Appears as *Block* not *Class* in many other contexts throughout Enterprise Architect

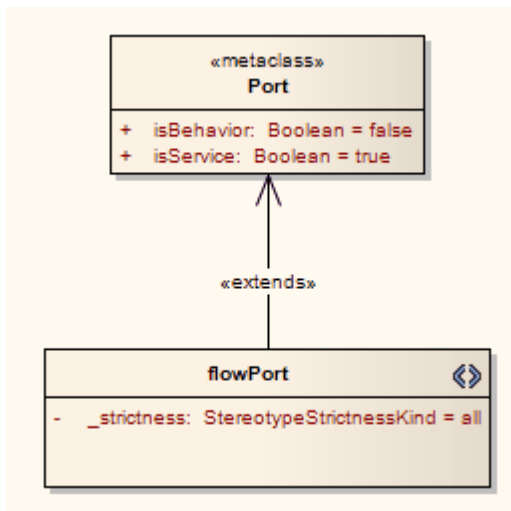
Learn more

- [Diagram](#) ⁵⁴⁹ (Strict UML Syntax Check)

The `_strictness` attribute is applied to a stereotype element. It defines to what level multiple stereotypes can be applied to an element. The type of the attribute is `StereotypeStrictnessKind` and it can have one of four values:

- *profile*, which states that an element of this type cannot be given more than one different stereotype from the same profile
- *technology*, which states that an element of this type cannot be given more than one different stereotype from the same technology
- *all*, which states that an element of this type cannot have multiple stereotypes at all, or
- *none*, which is the default Enterprise Architect behaviour and states that there are no restrictions on the use of multiple stereotypes

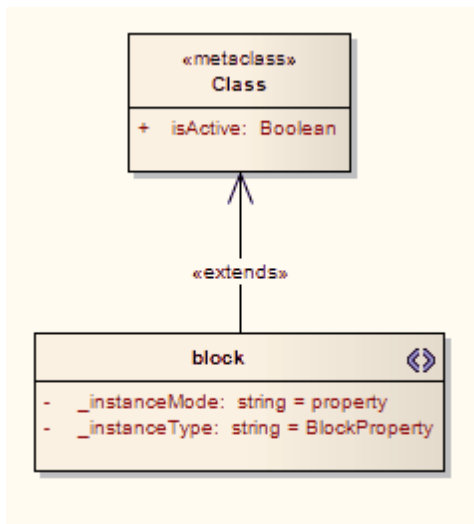
The following example is from SysML and shows that a `«flowPort»` cannot have any other stereotype applied to it.

Reference

Attribute	Meaning	See also
<code>_instanceMode</code>	Applied to a stereotype element, controlling the text in the Paste Element dialog after being translated. Valid values are instance and property , with the default being instance .	
<code>_instanceOwner</code>	Applied to a stereotype element, controlling the text in the Paste Element dialog. It is translated if it matches an Enterprise Architect element type; the default value	

Attribute	Meaning	See also
	is Element .	
_instanceType	Applied to a stereotype element, defining what kind of element is created as an instance of this element type. The value corresponds to the metatype given to a stereotype using the _metatype attribute. It is shown on the Paste Element dialog and is translated if it matches an Enterprise Architect element type.	

The following example from SysML shows that when an instance of a Block is created, it is created as a *BlockProperty* element.



Learn more

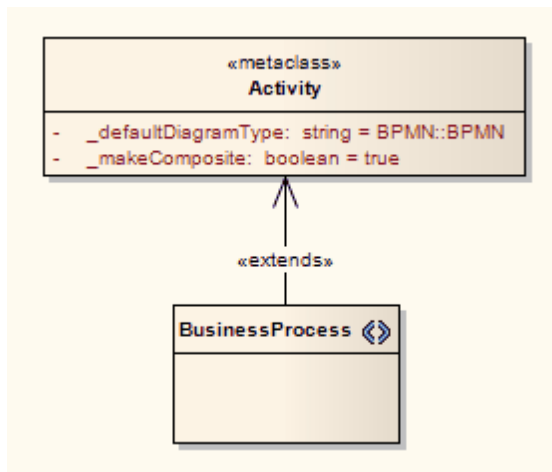
- [Paste Element from the Project Browser](#)^[733]

The **_makeComposite** attribute is applied to a metaclass element, not a stereotype element. It defines whether an element is always made composite when created.

Unless you also use the **_defaultDiagramType** attribute to define the child diagram type, the child diagram created is the usual default diagram type for the metaclass.

A stereotyped package is not by default created with a child diagram, so you should use the **_makeComposite** attribute to ensure the child diagram is created.

The following example from BPMN shows that a *BusinessProcess* element is always created as a Composite element with a BPMN custom child diagram.



Learn more

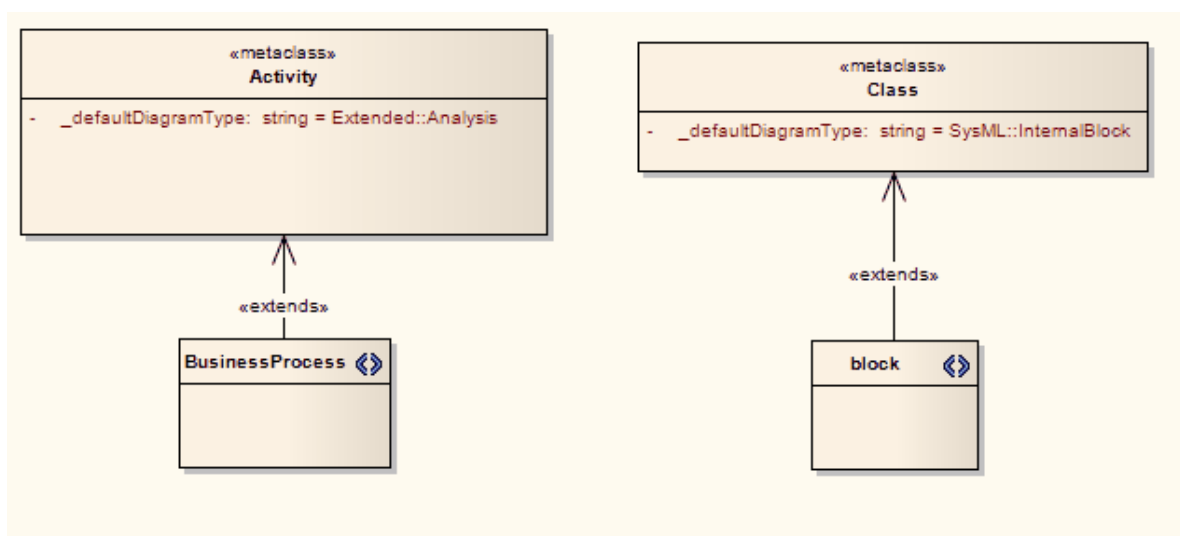
- [Defining Child Diagram Types](#) ¹³¹⁵

The **_defaultDiagramType** attribute is applied to a metaclass element, not a stereotype element. It defines the type of diagram created when an element is made composite.

This attribute can take as its name any of the inbuilt diagram types of Enterprise Architect as listed below.

Alternatively, if a custom diagram type is required, it should be prefixed with the *diagram profile name* and '::.'. The *diagram profile name* is the name given to the profile when you save it, which by default is the name of the profile package or profile diagram. If you follow the recommendation in *Create Diagram Profiles*, the diagram profile name is based on the technology name, but be aware that the attribute prefix is **not a direct** reference to the technology name.

The following examples show a «*BusinessProcess*»*Activity* that, when made a composite element, automatically creates an Analysis diagram, and a «*block*» stereotype that creates a SysML *InternalBlock* custom diagram.



You can also use the `_defaultDiagramType` attribute for packages, extending the Package metaclass.

Initial Values:

The following initial values for `_defaultDiagramType` should be used to refer to Enterprise Architect's inbuilt diagram types:

- UML Behavioral::Use Case
- UML Behavioral::Activity
- UML Behavioral::State Machine
- UML Behavioral::Communication
- UML Behavioral::Sequence
- UML Behavioral::Timing
- UML Behavioral::Interaction Overview
- UML Structural::Package
- UML Structural::Class
- UML Structural::Object
- UML Structural::Composite Structure
- UML Structural::Component
- UML Structural::Deployment
- Extended::Custom
- Extended::Requirements
- Extended::Maintenance
- Extended::Analysis
- Extended::User Interface
- Extended::Data Modeling
- Extended::ModelDocument.

Learn more

- [Creating Composite Elements](#)^[1314]
- [Custom Diagram Types](#)^[1346]

6.10.1.1.2.10 Stereotype Profiles

Customized stereotypes should be contained in one or more profiles. The stereotypes within each profile use the profile name as the namespace. You then add the profiles into an MDG Technology.

Create one or more packages with the «*profile*» stereotype, each package name being the namespace. Within each package create profile diagrams defining all the stereotypes in the namespace. You can use multiple diagrams to do this, but do not use nested packages.

Give the «*profile*» package a description in the **Notes** field, such as *MDG Technology for BPMN*. When all of the stereotypes are defined (make sure that every stereotype extends at least one *Metaclass*) right-click on the profile package in the Project Browser and select the **Save Package as UML Profile** context menu

option, then proceed as usual.

To define an MDG technology's Toolbox pages, you create a separate profile.

Learn more

- [Save Package as a UML Profile](#) ^[1306]
- [Add Profile to an MDG Technology](#) ^[1327]
- [Toolbox Profiles](#) ^[1337]

6.10.1.1.3 Quick Linker

Topics

Topic	Discussion	See also
Introduction	<p>The Quick Linker provides a fast and simple way to create new elements and connectors on a diagram.</p> <p>When an element is selected in a diagram, the Quick Linker arrow is displayed in the upper right corner of a element; simply clicking and dragging the arrow enables you to create new connectors and elements.</p> <p>The philosophy behind the built-in Quick Linker definitions is to provide, not a complete list of valid or legal connections, but a short and convenient list of the commonest connections for the given context.</p> <p>As part of a UML Profile, you can add to or replace the built-in Quick Linker definitions.</p>	<p>Quick Linker Definition Format ^[1318]</p> <p>Quick Linker Example ^[1320]</p> <p>Hide Default Quick Linker Settings ^[1321]</p> <p>Quick Linker Object Names ^[1322]</p>
Customized Quick Linker Settings	<p>A Quick Linker definition is a Comma Separated Value (CSV) format file; it is best manipulated in a spreadsheet, which should be set up to save the CSV file as comma-separated text without quotation marks around text fields.</p> <p>To add a Quick Linker definition file to a profile or technology, simply place a <i>DocumentArtifact</i> element onto the Profile diagram, give it the name <i>QuickLink</i>, then double-click on it.</p> <p>Open your CSV file in a text editor such as Notepad and copy and paste the contents into the <i>DocumentArtifact</i> element; the definitions are saved with the profile and are processed and applied when the profile is imported.</p> <p>The same applies if a profile is included within a technology, with the proviso that the QuickLink element must be in the same profile as the link stereotype definitions; this means that a technology could have a set of Quick Link definitions for each profile.</p>	

6.10.1.1.3.1 Quick Linker Definition Format

A Quick Linker definition is a text file consisting of records terminated by new-line characters. Each record must consist of 23 comma-separated fields, as defined by the table below. The values of each field must not be in quotes (" ").

A Quick Linker definition can include comments: all lines that begin with // are ignored by Enterprise Architect.

Each record of the Quick Linker definition represents a single entry on the Quick Linker menu. Some fields define the menu command; some fields can be thought of as filters, with the entry being ignored if the filter condition isn't met.

A Quick Linker definition has the following fields:

Column	Field	Description
A	Source Element Type	The row is ignored unless a connector is being dragged away from this type of element.
B	Source Stereotype Filter	If set, the row is ignored unless a connector is being dragged away from an element with this stereotype.
C	Target Element Type	If set, the row is ignored unless a connector is being dragged onto this type of element. If blank, the row is ignored unless a connector is being dragged onto an empty piece of diagram.
D	Target Stereotype Filter	If set and Target Element Type is also set, the row is ignored unless a connector is being dragged onto an element with this stereotype.
E	Diagram Filter	Contains either an inclusive or exclusive list of built-in diagrams ^[1348] , which limits the diagrams the given kind of connector can be included on. Each diagram name is terminated by a semi-colon; excluded diagram names are preceded by an exclamation mark. Example of an inclusive list: <i>Collaboration;Object;Custom;</i> Example of an Exclusive list: <i>!Sequence;</i>
F	New Element Type	If set and Create Element is also set, results in the creation of an element of this type.
G	New Element Stereotype	If set and Create Element is also set, results in the creation of an element with this stereotype.
H	New Link Type	If set and Create Link is also set, results in the creation of a connector of this type.

Column	Field	Description
I	New Link Stereotype	If set and Create Link is also set, results in the creation of a connector with this stereotype.
J	New Link Direction	<p>Can be:</p> <ul style="list-style-type: none"> • directed (always creates an association from source to target) • from (always creates an association from target to source) • undirected (always creates an association with unspecified direction) • bidirectional (always creates a bi-directional association), or • to (creates either a directed or undirected association, depending on the value of the Association Direction option) <p>Not all of these work with all connector types; for example, you cannot create a bi-directional Generalization.</p>
K	New Link Caption	If a new connector is being created but not a new element, then this is the text that appears on the context menu.
L	New Link & Element Caption	If a new connector AND a new element are being created, then this is the text that appears on the context menu.
M	Create Link	If set to TRUE , results in creation of a new connector; otherwise should be left blank.
N	Create Element	<p>If set to TRUE the row is ignored unless a connector is being dragged onto an empty piece of diagram and results in creation of a new element; otherwise should be left blank</p> <p>This overrides the values of Target Element Type and Target Stereotype Filter.</p>
O	Disallow Self connector	Should be set to TRUE if self connectors are invalid for this kind of connector; otherwise should be left blank.
P	Exclusive to ST Filter + No inherit from Metatype	If set to TRUE , indicates that elements of type <i>Source Element Type</i> with the stereotype <i>Source Stereotype Filter</i> do not display the Quick Linker definitions of the equivalent unstereotyped element.
Q	Menu Group	If set, indicates the name of a sub-menu in which a menu item is created.

Column	Field	Description
R	Complexity Level	Not implemented, always set to 0 .
S	Target Must Be Parent	If set to TRUE this menu item only appears when dragging from a child element to its parent; for example from a port to its containing Class.
T	Embed element	If set to TRUE the element being created is embedded in the target element; otherwise should be left blank.
U	Precedes Separator LEAF	If set to TRUE results in a menu separator being added to the Quick Linker menu; otherwise should be left blank.
V	Precedes Separator GROUP	If set to TRUE results in a menu separator being added to the Quick Linker sub-menu; otherwise should be left blank.
W	Dummy Column	Depending on which spreadsheet application you use, this column might require a value in every cell to force CSV export to work correctly with trailing blank values.

6.10.1.1.3.2 Quick Linker Example

This example uses a Class element with the stereotype «*quick*». The example scenario is this: when you drag a connector away from one of these elements, you want to create a Dependency either to or from a component element. When you drag a connector onto an existing *Port* or component element, you want a Dependency either to or from the component **or**, in the case of a component, you want to be able to create an embedded Port element.

This results in 8 records in the [Quick Linker definition](#)  file.

1. Dependency to new Component
2. Dependency from new Component
3. Dependency to existing Component
4. Dependency from existing Component
5. Dependency to existing Port
6. Dependency from existing Port
7. Dependency to existing Component, create new Port
8. Dependency from existing Component, create new Port

In the spreadsheet, this is implemented by the following values:

	A	B	C	E	F	H	J	K
1	//Source Element Type	Source ST filter	Target Element Type	Diagram Filter	New Element Type	New Link Type	New Link Direction	New Link Caption
2	Class	quick			Component	Dependency	to	
3	Class	quick			Component	Dependency	from	
4	Class	quick	Component			Dependency	to	Dependency to
5	Class	quick	Component			Dependency	from	Dependency from
6	Class	quick	Port			Dependency	to	Dependency to
7	Class	quick	Port			Dependency	from	Dependency from
8	Class	quick	Component		Port	Dependency	to	
9	Class	quick	Component		Port	Dependency	from	

	L	M	N	O	P	Q	R	S	T	U	V
1	New Link & Element Caption	Create Link	Create Element	Disallow Self connector	No inherit from metatype	Menu Group	Complexity Level	Target Must Be Parent	Embed element	Precedes Separator LEAF	Precedes Separator GROUP
2	Dependency to	TRUE	TRUE	TRUE	TRUE	Component	0				
3	Dependency from	TRUE	TRUE	TRUE	TRUE	Component	0			TRUE	
4		TRUE		TRUE	TRUE		0				
5		TRUE		TRUE	TRUE		0			TRUE	
6		TRUE		TRUE	TRUE		0				
7		TRUE		TRUE	TRUE		0			TRUE	
8	Dependency to	TRUE	TRUE	TRUE	TRUE	Port	0		TRUE		
9	Dependency from	TRUE	TRUE	TRUE	TRUE	Port	0		TRUE	TRUE	

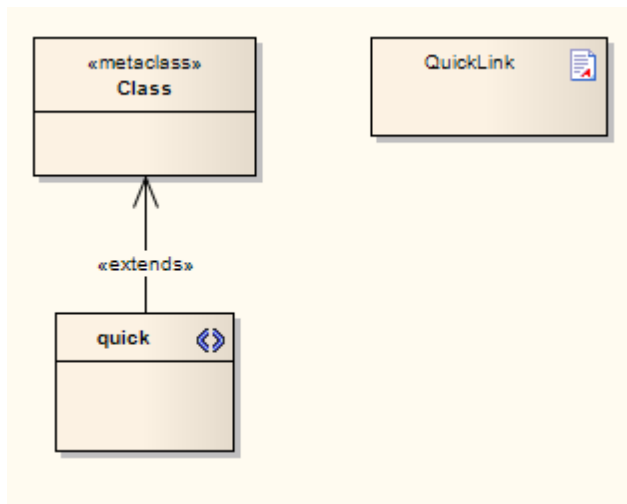
This saves to the following CSV:

```

Class,quick,,,,Component,,Dependency,,to,,Dependency to,TRUE,TRUE,TRUE,
TRUE,Component,0,,,,,
Class,quick,,,,Component,,Dependency,,from,,Dependency from,TRUE,TRUE,
TRUE,TRUE,Component,0,,,TRUE,,
Class,quick,Component,,,,,Dependency,,to,Dependency to,,TRUE,,TRUE,
TRUE,,0,,,,,
Class,quick,Component,,,,,Dependency,,from,Dependency from,,TRUE,,TRUE,
TRUE,,0,,,TRUE,,
Class,quick,Port,,,,,Dependency,,to,Dependency to,,TRUE,,TRUE,
TRUE,,0,,,,,
Class,quick,Port,,,,,Dependency,,from,Dependency from,,TRUE,,TRUE,
TRUE,,0,,,TRUE,,
Class,quick,Component,,,Port,,Dependency,,to,,Dependency to,TRUE,TRUE,
TRUE,TRUE,Port,0,,TRUE,,,
Class,quick,Component,,,Port,,Dependency,,from,,Dependency from,TRUE,
TRUE,TRUE,TRUE,Port,0,,TRUE,TRUE,,

```

You can create the following profile and cut and paste the CSV data into the document artifact to test the effect.



6.10.1.1.3.3 Hide Default Quick Linker Settings

If you have a Quick Linker definition with the *Exclusive to stereotype* flag (column P) set to **TRUE**, then the default Quick Linker definitions between the given source and target are overridden. However, you might want to override the defaults without actually having a Quick Linker definition. For example, if you don't define any Quick Links for a «quick» Class to another «quick» Class, Enterprise Architect displays the default Quick Links for a Class to another Class.

To override this behaviour, create a Quick Linker definition that has the source element type, source stereotype filter, target element type and target stereotype filter fields (columns A, B, C and D) all set, with the **Exclusive to stereotype** flag (column P) set to **TRUE**, and with the new link type field (column H) set to **<none>**.

For example, add this line to the example in [Quick Linker Example](#)^[1320]:

```
Class, quick, Interface, , , , <none>, , , , , TRUE, , 0, , , ,
```

This overrides the default Class-to-Interface Quick Links when a Quick Link is dragged from a «quick» Class to an Interface element.

This technique does not affect the automatic appearance of Dependency, Trace, Information Flow and Help items on the Quick Linker menu.

6.10.1.1.3.4 Quick Linker Object Names

List of Element Types:

The following element names can be used in Quick Linker definitions:

Action	ExecutionEnvironment	Package
ActionPin	ExitPoint	Part
Activity	ExitState	Port
ActivityParameter	ExpansionNode	PrimitiveType
ActivityPartition	ExpansionRegion	ProvidedInterface
Actor	Feature	Receive

Artifact	GUIElement	RequiredInterface
Boundary	HistoryState	Requirement
CentralBufferNode	InformationItem	Screen
Change	InitialActivity	Send
ChoiceState	InitialState	Sequence
Class	InteractionOccurrence	Signal
Collaboration	Interface	State
Component	Issue	StateLifeline
DataType	InterruptableActivityRegion	StateMachine
Decision	JunctionState	Synchronization_H
DeepHistoryState	MergeNode	Synchronization_V
Deployment Specification	MessageEndpoint	SynchState
Device	n-ary Association	UMLDiagram
DiagramGate	Node	UseCase
EntryPoint	Object	ValueLifeline
EntryState	ObjectNode	

List of Connector Types:

The following connector names can be used in Quick Linker definitions:

Aggregation	Deployment	PackageMerge
Association	Extension	Realization
AssociationClass	Generalization	Redefinition
CommunicationPath	InformationFlow	Sequence
Composition	InterfaceLink	StateFlow
ConnectorLink	Manifest	TemplateBinding
ControlFlow	Nesting	UCExtends
DelegateLink	ObjectFlow	UCIncludes
Dependency	PackageImport	UseCase

6.10.1.2 MDG Technologies - Creating

The Model Driven Generation (MDG) Technologies enable Enterprise Architect users to access and use resources pertaining to a specific technology in Enterprise Architect. There are various options for an administrator or individual user to bring MDG Technologies into use with Enterprise Architect. You should read the first two MDG Technology topics (below) to understand how MDG Technologies are accessed and used within Enterprise Architect.

A further option is that Technology Developers can develop new MDG Technologies and deploy them to the project team as appropriate, as described in the remaining topics listed below.

Learn more

- [MDG Technologies](#) ^[1283]
- [Manage MDG Technologies](#) ^[1285]
- [Create MDG Technologies](#) ^[1324]
- [Working With MTS Files](#) ^[1335]
- [Customize Toolbox Profiles](#) ^[1337]
- [Create Diagram Profiles](#) ^[1346]
- [Create Learning Center Profiles](#) ^[1350]
- [Define Validation Configuration](#) ^[1354]
- [Incorporate Model Templates](#) ^[1355]
- [Deploy an MDG Technology](#) ^[1357]

An example of creating an MDG Technology for an Enterprise Architecture framework is provided in: [Enterprise Architecture Framework Design with Sparx Systems Enterprise Architect](#).

[Set Up Technology Element Images](#) ^[1356] describes the process of ensuring that a technology element is represented by a specific image on a diagram, when the technology is deployed in a model.

6.10.1.2.1 Create MDG Technologies

Using the MDG Technology Wizard, you can create MDG Technology files that can include UML Profiles, code modules, Patterns, images, Tagged Value Types, RTF report templates, linked document templates, Toolbox pages and Learning Center pages.

This facility is not available in the Enterprise Architect Desktop edition.

Access [Tools | Generate MDG Technology File](#)

How to

To create an MDG Technology file

Step	Description	See also
1	Select the Generate MDG Technology File menu option. The MDG Technology Creation Wizard screen displays.	
2	Click on the Next button. The MDG Technology Wizard prompts you to either: <ul style="list-style-type: none"> • Create an MDG Technology File by creating a new MDG Technology Selection (MTS) file • Create an MDG Technology File using an existing MTS file, or 	

Step	Description	See also														
	<ul style="list-style-type: none"> • Not use any MTS file <p>(An MTS file stores the selected options that you define during the creation of an MDG Technology File; if you use an MTS file, you can later modify it to add or remove specific items in the MDG Technology File, which is the recommended process.)</p>															
3	<p>Select the appropriate MTS file option.</p> <p>Click on the Next button.</p> <p>If you selected an MTS file, the MDG Technology Wizard prompts you to save the changes in the existing MTS file or into a new MTS file; this enables you to create a modification based on the existing MTS file, while preserving the original file.</p>															
4	<p>If necessary, type in or browse for the required file path and name.</p> <p>Click on the Next button.</p> <p>The MDG Technology Wizard - Create screen displays.</p>															
5	<p>Complete the fields on this screen as follows:</p> <table border="1" data-bbox="335 1155 1198 2042"> <thead> <tr> <th data-bbox="335 1155 507 1211">Option</th> <th data-bbox="507 1155 1198 1211">Action</th> </tr> </thead> <tbody> <tr> <td data-bbox="335 1211 507 1317">Technology</td> <td data-bbox="507 1211 1198 1317">Type the name of the MDG Technology.</td> </tr> <tr> <td data-bbox="335 1317 507 1462">Filename</td> <td data-bbox="507 1317 1198 1462">Type or select the path and filename of the MDG Technology File. The file extension for this file is .xml.</td> </tr> <tr> <td data-bbox="335 1462 507 1592">ID</td> <td data-bbox="507 1462 1198 1592">Type a reference for the MDG Technology File, up to 12 characters long.</td> </tr> <tr> <td data-bbox="335 1592 507 1697">Version</td> <td data-bbox="507 1592 1198 1697">Type the version number of the MDG Technology File.</td> </tr> <tr> <td data-bbox="335 1697 507 1899">Icon</td> <td data-bbox="507 1697 1198 1899">(Optional) Type or select the path and file name of the graphics containing the technology icon. The icon is a 16x16 bitmap image that is shown in the list of technologies on the left of the MDG Technologies dialog.</td> </tr> <tr> <td data-bbox="335 1899 507 2042">Logo</td> <td data-bbox="507 1899 1198 2042">(Optional) Type or select the path and file name of the graphics containing the technology logo. The logo is a 64x64 bitmap image that is shown in the display</td> </tr> </tbody> </table>	Option	Action	Technology	Type the name of the MDG Technology.	Filename	Type or select the path and filename of the MDG Technology File. The file extension for this file is .xml.	ID	Type a reference for the MDG Technology File, up to 12 characters long.	Version	Type the version number of the MDG Technology File.	Icon	(Optional) Type or select the path and file name of the graphics containing the technology icon. The icon is a 16x16 bitmap image that is shown in the list of technologies on the left of the MDG Technologies dialog.	Logo	(Optional) Type or select the path and file name of the graphics containing the technology logo. The logo is a 64x64 bitmap image that is shown in the display	
Option	Action															
Technology	Type the name of the MDG Technology.															
Filename	Type or select the path and filename of the MDG Technology File. The file extension for this file is .xml.															
ID	Type a reference for the MDG Technology File, up to 12 characters long.															
Version	Type the version number of the MDG Technology File.															
Icon	(Optional) Type or select the path and file name of the graphics containing the technology icon. The icon is a 16x16 bitmap image that is shown in the list of technologies on the left of the MDG Technologies dialog.															
Logo	(Optional) Type or select the path and file name of the graphics containing the technology logo. The logo is a 64x64 bitmap image that is shown in the display															

Step	Description	See also										
	<table border="1"> <thead> <tr> <th>Option</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td></td> <td>on the top-right corner of the MDG Technologies dialog.</td> </tr> <tr> <td>URL</td> <td>(Optional) If you have any website product information that might be helpful for users of this Technology, type or paste the URL in the field.</td> </tr> <tr> <td>Support</td> <td>(Optional) If you have any web-based or other support facility that might be helpful for users of this Technology, type or paste the contact address in this field.</td> </tr> <tr> <td>Notes</td> <td>Type a short explanation of the functionality of the MDG Techno</td> </tr> </tbody> </table>	Option	Action		on the top-right corner of the MDG Technologies dialog.	URL	(Optional) If you have any website product information that might be helpful for users of this Technology, type or paste the URL in the field.	Support	(Optional) If you have any web-based or other support facility that might be helpful for users of this Technology, type or paste the contact address in this field.	Notes	Type a short explanation of the functionality of the MDG Techno	
Option	Action											
	on the top-right corner of the MDG Technologies dialog.											
URL	(Optional) If you have any website product information that might be helpful for users of this Technology, type or paste the URL in the field.											
Support	(Optional) If you have any web-based or other support facility that might be helpful for users of this Technology, type or paste the contact address in this field.											
Notes	Type a short explanation of the functionality of the MDG Techno											
6	Click on the Next button. The MDG Technology Wizard - Contents screen displays.											
7	Select the checkbox for each item to be included in the MDG Technology file. Each selection runs specific dialogs to enable definition of the specific items to be included in the MDG Technology, as described in the topics listed on the right.	Add a Profile ^[1327] Add a Pattern ^[1328] Add a Diagram Profile ^[1329] Add a Toolbox Profile ^[1329] Add Learning Center Pages ^[1330] Add Tagged Value Types ^[1331] Add Code Modules ^[1331] Add MDA Transformations ^[1333] RTF Report Templates ^[1334] Linked Document Templates ^[1335] Add Images ^[1333] Add Scripts ^[1334] (Corporate and 'Suite' editions)										
8	Work through the dialogs displayed in response to your choices, and when all											

Step	Description	See also
	are complete, click on the Next button. The MDG Technology Wizard - Finish screen displays, providing information on the items included in the MDG Technology File.	
9	If you have used an MTS file and want to update it, select the Save to MTS checkbox.	
10	If you are satisfied with the selection of items, click on the Finish button. You can now edit the MTS file, if required, to add further items such as: <ul style="list-style-type: none"> • Model Search definitions • Model Views • Model Validation configurations • Model Templates To make the MDG Technology File accessible to an Enterprise Architect model, you must <i>add</i> the technology file path to the MDG Technologies - Advanced dialog.	Working with MTS Files ^[1335] Access Remote MDG Technologies ^[1287]

6.10.1.2.1 Add a Profile

When creating an MDG Technology file, you can include UML 2.3-compliant profiles that you have defined.

To use the Profiles section of the MDG Technology Wizard

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Profiles checkbox. The MDG Technology Wizard - Profile files selection dialog displays.	Creating MDG Technologies. Step 6 ^[1326]
2	In the Directory field, navigate to the directory containing the required Profile or Profiles. The Profile files are automatically listed in the Available Files panel.	
3	To select each required Profile individually, highlight the Profile in the Available Files list and click on the --> button. The file name displays in the Selected Files list. <i>Alternatively:</i> To select all available Profiles, click on the -->> button. <ul style="list-style-type: none"> • DO NOT select diagram profiles or toolbox profiles on this dialog; this would generate conflicting commands in the .MTS file 	Stereotype Profile ^[1316]

Step	Description	See also
	<ul style="list-style-type: none"> Make sure you do include your stereotype profile 	
4	Click on the Next button to proceed.	

Learn more

- [Working With Profiles](#) ^[1295]

6.10.1.2.1.2 Add a Pattern

When creating an MDG Technology file, you can include patterns.

To use the Patterns section of the MDG Technology Wizard

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Patterns checkbox. The MDG Technology Wizard - Pattern files selection dialog displays.	Creating MDG Technologies, Step 6 ^[1326]
2	In the Directory field, navigate to the directory containing the required pattern or patterns. The pattern files are automatically listed in the Available Files panel.	
3	To select each required pattern individually, highlight the pattern in the Available Files list and click on the --> button. The file name displays in the Selected Files list. Alternatively, to select all available patterns, click on the -->> button.	
4	Click on the Next button to proceed.	

Learn More:

- [Creating Patterns](#) ^[1271]

6.10.1.2.1.3 Add a Diagram Profile

When creating an MDG Technology file, you can include a diagram profile that you have defined.

To use the diagram profiles section of the MDG Technology Wizard

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Diagram Types checkbox. The MDG Technology Wizard - Diagram Types dialog displays.	Creating MDG Technologies, Step 6 <small>1326</small>
2	In the Directory field, navigate to the directory containing the required diagram profiles. The profiles in the directory are automatically listed in the Available Files panel.	
3	To select each required diagram profile individually, highlight the file name in the Available Files list and click on the --> button. The file name displays in the Selected Files list. Alternatively, to select all available profiles (assuming they are all diagram profiles), click on the -->> button.	
4	Click on the Next button to proceed.	

Learn more

- [Custom Diagram Types](#)
1346

6.10.1.2.1.4 Add a Toolbox Profile

When creating an MDG Technology file, you can include Enterprise Architect Toolbox page definitions that you have created.

To use the Toolboxes section of the MDG Technology Wizard

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Toolboxes checkbox. The MDG Technology Wizard - Toolboxes dialog displays.	Creating MDG Technologies, Step 6 <small>1326</small>
2	In the Directory field, navigate to the directory containing the required toolbox profiles. The profile files are automatically listed in the Available Files panel.	

Step	Description	See also
3	To select each required toolbox profile individually, highlight the file name in the Available Files list and click on the --> button. The file name displays in the Selected Files list. Alternatively, to select all available profiles (assuming they are all toolbox profiles), click on the -->> button.	
4	Click on the Next button to proceed.	

Learn more

- [Custom Toolboxes](#)^[1337]

6.10.1.2.1.5 Add Learning Center Pages

When creating an MDG Technology file, you can include Enterprise Architect Learning Center profiles that you have created.

How to

To use the Taskpages section of the MDG Technology Wizard

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Taskpages checkbox. The MDG Technology Wizard - Learning Center dialog displays.	Creating MDG Technologies, Step 6 ^[1326]
2	In the Directory field, navigate to the directory containing the required taskpage profiles. The profile files are automatically listed in the Available Files panel.	
3	To select each required Learning Center profile individually, highlight the file name in the Available Files list and click on the --> button. The file name displays in the Selected Files list. Alternatively, to select all available profiles (assuming they are all Learning Center profiles), click on the -->> button.	
4	Click on the Next button to proceed.	

Learn more

- [Custom Learning Center](#)^[1350]

6.10.1.2.1.6 Add Tagged Value Types

When creating an MDG Technology file, you can include Tagged Value Types.

To use the Tagged Value Types section of the MDG Technology Types Wizard

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Tagged Value Types checkbox. The MDG Technology Wizard - Tagged Value Types dialog displays.	Creating MDG Technologies, Step 6 ^[1326]
2	To select each required Tagged Value Type individually, highlight the file name in the Available Files list and click on the --> button. The file name displays in the Selected Files list. Alternatively, to select all available Tagged Value Types, click on the -->> button.	
3	Click on the Next button to proceed.	

Learn more

- [Tagged Value Types](#) ^[1385]

6.10.1.2.1.7 Add Code Modules

When creating an MDG Technology file, you can include code modules.

To use the code modules section of the MDG Technology Types Wizard

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Code Modules checkbox. The MDG Technology Wizard - Code Modules dialog displays.	Creating MDG Technologies, Step 6 ^[1326]
2	Click on the checkboxes (Product , Data Types , and Code Templates) for each of the required Code Modules. The code modules listed are those defined in your current project. These could be the Enterprise Architect default languages, or those you have defined yourself using code templates and the Code Template Editor. Before you can set up a code template for the new language in the editor, you must define at least one data type for the language. Once the MDG Technology file is created it can be loaded into your current model and into other models.	Code and Transform Templates ^[1395] Code Template Editor ^[1404] Data Types ^[987]
3	To select any code options for a module, click on the (...) button in the Code Options column for that module. This enables you to select an XML document	

Step	Description	See also																				
	<p>that provides additional settings for the language that are not covered by the data types or code templates.</p> <p>The root node of the XML document should be <code>CodeOptions</code>. The child nodes should be called <code>CodeOption</code> and should contain a <i>name</i> attribute.</p> <p>The supported code options are as follows:</p> <table border="1" data-bbox="331 555 1209 1709"> <thead> <tr> <th data-bbox="331 555 651 611">Code Option</th> <th data-bbox="651 555 1209 611">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="331 611 651 745">ConstructorName</td> <td data-bbox="651 611 1209 745">The name of a function used as a constructor. Used with the <code>classHasConstructor</code> code template macro.</td> </tr> <tr> <td data-bbox="331 745 651 880">CopyConstructorName</td> <td data-bbox="651 745 1209 880">The name of a function used as a copy constructor. Used with the <code>classHasCopyConstructor</code> code template macro.</td> </tr> <tr> <td data-bbox="331 880 651 981">DefaultExtension</td> <td data-bbox="651 880 1209 981">The default extension when generating code.</td> </tr> <tr> <td data-bbox="331 981 651 1081">DefaultSourceDirectory</td> <td data-bbox="651 981 1209 1081">The default path to which Enterprise Architect generates code.</td> </tr> <tr> <td data-bbox="331 1081 651 1216">DestructorName</td> <td data-bbox="651 1081 1209 1216">The name of a function used as a destructor. Used with the <code>classHasDestructor</code> code template macro.</td> </tr> <tr> <td data-bbox="331 1216 651 1317">Editor</td> <td data-bbox="651 1216 1209 1317">The external editor used for editing source of this language.</td> </tr> <tr> <td data-bbox="331 1317 651 1440">ImplementationExtension</td> <td data-bbox="651 1317 1209 1440">The extension used by Enterprise Architect to generate implementation file.</td> </tr> <tr> <td data-bbox="331 1440 651 1574">ImplementationPath</td> <td data-bbox="651 1440 1209 1574">The relative path from the source file to generate the implementation file.</td> </tr> <tr> <td data-bbox="331 1574 651 1709">PackagePathSeparator</td> <td data-bbox="651 1574 1209 1709">The delimiter used to separate package names with the <code>packagePath</code> macro from the code templates.</td> </tr> </tbody> </table> <p>An example of a valid code options file is shown below.</p> <pre data-bbox="419 1798 1153 1955"> <CodeOptions> <CodeOption name="DefaultExtension">.ext</CodeOption> <CodeOption name="Editor">C:\Windows\notepad.exe</CodeOption> </CodeOptions> </pre>	Code Option	Description	ConstructorName	The name of a function used as a constructor. Used with the <code>classHasConstructor</code> code template macro.	CopyConstructorName	The name of a function used as a copy constructor. Used with the <code>classHasCopyConstructor</code> code template macro.	DefaultExtension	The default extension when generating code.	DefaultSourceDirectory	The default path to which Enterprise Architect generates code.	DestructorName	The name of a function used as a destructor. Used with the <code>classHasDestructor</code> code template macro.	Editor	The external editor used for editing source of this language.	ImplementationExtension	The extension used by Enterprise Architect to generate implementation file.	ImplementationPath	The relative path from the source file to generate the implementation file.	PackagePathSeparator	The delimiter used to separate package names with the <code>packagePath</code> macro from the code templates.	<p>Field Substitution Macros <small>1417</small></p>
Code Option	Description																					
ConstructorName	The name of a function used as a constructor. Used with the <code>classHasConstructor</code> code template macro.																					
CopyConstructorName	The name of a function used as a copy constructor. Used with the <code>classHasCopyConstructor</code> code template macro.																					
DefaultExtension	The default extension when generating code.																					
DefaultSourceDirectory	The default path to which Enterprise Architect generates code.																					
DestructorName	The name of a function used as a destructor. Used with the <code>classHasDestructor</code> code template macro.																					
Editor	The external editor used for editing source of this language.																					
ImplementationExtension	The extension used by Enterprise Architect to generate implementation file.																					
ImplementationPath	The relative path from the source file to generate the implementation file.																					
PackagePathSeparator	The delimiter used to separate package names with the <code>packagePath</code> macro from the code templates.																					

Step	Description	See also
4	Click on the Next button to proceed. You can edit the code option values for source code engineering and for each required language using the appropriate Language Options page of the Options dialog.	Code Engineering Settings [1897] Language Options [1907]

6.10.1.2.1.8 Add MDA Transforms

When creating an MDG Technology file, you can include the MDA Transformations that have been modified in the model.

To use the Transform Modules section of the MDG Technology Wizard

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the MDA Transforms checkbox. The MDG Technology Wizard - Transform Modules dialog displays.	Creating MDG Technologies, Step 6 [1326]
2	Click the checkbox against the template name of each required template that is present in the current model.	
3	Click on the Next button to proceed.	

6.10.1.2.1.9 Add Images

When creating an MDG Technology file, you incorporate images to be used in all models in which the technology is *deployed*. These images must already be available in the model in which the technology is being *developed*; you can import the images into this model using the **Add New** button on the Image Manager.

How to

To use the Image Selection section of the MDG Technology Wizard

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Images checkbox. The MDG Technology Wizard - Image Selection dialog displays.	Creating MDG Technologies, Step 6 [1326]
2	For each required model image available in the current model, select the checkbox next to the image name. A preview of each image displays on the right of the dialog as you select the	

Step	Description	See also
	checkbox.	
3	Click on the Next button to proceed.	

Learn more

- [Image Manager](#)^[755]

6.10.1.2.1.10 Add Scripts

When creating an MDG Technology file, you can include scripts that you have created in the model.

To use the Script Selection section of the MDG Technology Wizard

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Scripts checkbox. The MDG Technology Wizard - Scripts dialog displays.	Creating MDG Technologies. Step 6 ^[1326]
2	For each required script available in the current model, select the checkbox next to the script name.	
3	Click on the Next button to proceed.	

Notes

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

Learn more

- [The Scripter Window](#)^[2372]

6.10.1.2.1.11 Add RTF Report Templates

When creating an MDG Technology file, you can include user-defined RTF Report templates.

To use the report templates section of the MDG Technology Wizard

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the RTF Templates checkbox.	Creating MDG Technologies. Step 6 ^[1326]

Step	Description	See also
	The MDG Technology Wizard - RTF Report Templates dialog displays.	
2	For each required user-defined report template available in the current model, select the checkbox next to the template name.	
3	Click on the Next button to proceed.	

6.10.1.2.1.12 Add Linked Document Templates

When creating an MDG Technology file, you can include Linked Document templates.

To use the Linked Document templates section of the MDG Technology Wizard

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Linked Document Templates checkbox. The MDG Technology Wizard - Linked Document Templates dialog displays.	Creating MDG Technologies, Step 6 <small>[1326]</small>
2	For each required document template available in the current model, select the checkbox next to the template name.	
3	Click on the Next button to proceed.	

6.10.1.2.2 Working with MTS Files

An MDG Technology Selection (.MTS) file stores the selected options that you define when creating an MDG Technology File using the MDG Technology Wizard.

If you use a .MTS file, you can edit it to change the features selected when you generated the file, and to add or remove the advanced features described in this topic.

Topics

Action	Description	See also
Create a .MTS File	To create a .MTS file, select the Tools Generate MDG Technology File menu option to launch the MDG Technology Wizard, and work through the screens as described in the <i>Create MDG Technologies</i> topic.	Create MDG Technologies <small>[1324]</small>

Action	Description	See also
	On the second page, select the Create a new MTS file option.	
Advanced Options For Your .MTS File	<p>Once you have created the .MTS file, you can add:</p> <ul style="list-style-type: none"> • Model Search definitions - If you use a custom SQL search, the SQL must include <i>ea_guid AS CLASSGUID</i> and the <i>object type</i> • Model Views - Technology views do not store Favorite packages, only Views; if your exported views run searches that you have defined you must also include those searches in your MDG Technology • Model Validation configurations • Model Templates <p>Open the .MTS file in a text editor; to make it easier, copy the following lines and paste them into the file before the last line of the file (that is, just before the <code></ MDG. S e l e c t i o n s ></code> lines:</p> <pre><Model Searches file="" / > <Model Views file="" / ></pre> <p>(The code for the model validation configurations and model templates is provided in the corresponding sections, accessed via the links in the right-hand column.)</p> <p>You can, if necessary, have more than one line for each inclusion - for example, more than one <i>ModelSearch</i>; for each inserted line:</p> <ul style="list-style-type: none"> • In the <code>f i l e</code> attribute, enter the filename of the Model Search XML file or Model View XML file <p>Save the .MTS file.</p>	Model Search Definitions ^[621] Model Views ^[604] Model Validation Configurations ^[1354] Model Templates ^[1355]
Update the MDG Technology	<p>Again select the Tools Generate MDG Technology File menu option, but this time on the second page select Open an Existing MTS file and specify the file path of the .MTS file you have updated.</p> <p>Click on <Next> until the wizard is finished; your MDG Technology file is updated.</p>	

6.10.1.2.3 Customize Toolbox Profiles

The following is a road map of how to create a set of custom toolboxes for Enterprise Architect.

Step	Action	See also
1	Create a set of Toolbox Profiles that contain the definitions that Enterprise Architect requires to create the toolboxes. These might contain scripts to include customized compartments to your element images.	Toolbox Profiles ^[1337] Add Custom Compartments to Element ^[1339]
2	Create a .MTS file containing instructions on how to build your MDG Technology; use this .MTS file to build your MDG Technology.	Creating MDG Technologies ^[1324]
3	Add some finishing touches: <ul style="list-style-type: none"> • Create hidden sub-menus • Override Enterprise Architect's default toolboxes • Change the default icons for toolbox items 	Hidden Sub-menus ^[1341] Overriding Default Toolboxes ^[1342] Icons for Toolbox Items ^[1342]

6.10.1.2.3.1 Create Toolbox Profiles

You can create multiple toolbox profiles within an MDG Technology. Each toolbox profile contains definitions that determine what appears in the Toolbox when it is opened, either by selecting from the **More tools...** option in the Toolbox window, or by opening or creating a diagram of the type that is linked to the toolbox profile.

How to

To create a toolbox profile

Step	Action	See also
1	Create a diagram in a profile package, and give it a name by which you can refer to it later, such as <i>MyClassDiagram</i> .	
2	In the Notes field, give the diagram an alias and a description in the following format: <pre>Alias=MyClass; Notes=Structural elements for class diagrams;</pre>	
3	On the diagram, create a Class, name it <i>ToolboxPage</i> and give it the «metaclass» stereotype.	
4	Create a «stereotype» element for each of the toolbox pages to create within your toolbox, such as <i>MyClassElements</i> and <i>MyClassRelationships</i> .	Toolbox Page Attributes ^[1339]

Step	Action	See also
	<p>Set their Alias to the text to display in the title bar of each toolbox page, such as <i>My Class Elements</i> and <i>My Class Relationships</i> respectively.</p> <p>Use the Notes field to define the tool-tip for each toolbox page; that is, Elements for Class Diagrams and Relationships for Class Diagrams.</p> <p>Use the «<i>extends</i>» connector to set the stereotype elements to extend <i>ToolboxPage</i>.</p>	
5	<p>In the «<i>stereotype</i>» elements, create an attribute for each toolbox item.</p> <p>The name of the attribute should be the name of the element or connector to be dropped, including namespace; for example, <i>UML::Package</i>, <i>UML::Class</i> and <i>UML::Interface</i>.</p> <p>The toolbox items display in the same order as the attributes in the Class, so make use of the attribute ordering buttons to define the order of icons in your toolbox page. You might also include custom compartments in your element images.</p> <p>To name an attribute for an item from your own technology, precede it with your profile name as the namespace, and then follow it in brackets with the element or connector type that you are extending (so that Enterprise Architect knows what object to create); for example, a SysML block element would appear as <i>SysML::Block(UML::Class)</i>.</p> <p>Click on the links on the right for a complete list of elements and connectors that can be extended.</p> <p>To define a toolbox item that allows a pattern to be dropped onto a diagram, name the attribute <i>My Technology::MyPattern(UMLPattern)</i> where <i>MyTechnology</i> is the ID of the technology and <i>MyPattern</i> is the name of the pattern to drop; for example, <i>BusFramework::Builder(UMLPattern)</i>.</p> <p>You can replace (<i>UMLPattern</i>) with (<i>UMLPatternSilent</i>) to avoid displaying the Add Pattern dialog.</p> <p>To define a model-based pattern in a custom toolbox (such as the GoF patterns) create an attribute with a name of the format <i>PatternCategory::PatternName(UMLPattern)</i>; for example: <i>GoF Behavioral Patterns:: Mediator(UML Pattern)</i>.</p> <p>You might not want to use names such as <i>UML::Package</i> or <i>UML::Class</i> in your toolbox, so give the attributes an Initial Value of, for example, <i>Package</i> or <i>Class</i>.</p>	<p>Add Custom Compartments to Element^[1339]</p> <p>Elements Used in Toolboxes^[1340]</p> <p>Connectors Used in Toolboxes^[1341]</p>
6	<p>To save the toolbox profile, right-click on the diagram and select the Save as Profile context menu option.</p>	

Notes

- Each profile element incorporated into an MDG Toolbox page enables synchronization of the Tagged Values and Constraints of all elements created from them
- The sequence of Toolbox pages in the Toolbox is determined by the sequence of their stereotyped Class elements in the profile diagram or profile package; if you create and save the profile from a:
 - Diagram, the Toolbox page sequence is determined by the Z-order of the stereotyped Class

elements on the diagram - the higher the z-order number of the stereotyped Class element, the further down the Toolbox its Toolbox page is placed; if you change the Z-order of a stereotyped element in the diagram, it changes the position of the element's page on the Toolbox

- Package in the Project Browser, the Toolbox page sequence is determined by the list order of the stereotyped Class elements in the package - the Toolbox page for the first listed element is at the top of the Toolbox; if you re-order the elements in the Project Browser list, you produce the same re-ordering of pages in the Toolbox

Learn more

- [Synchronize Tags and Constraints](#) ^[1281]
- [Z Order Elements](#) ^[744]
- [Diagram Context Menu](#) ^[687] (**Modify Z-Order** option)

The following attributes can be added to a stereotype Class that extends the *ToolboxPage* metaclass:


- **ImagesOnly**: if you give a toolbox page an attribute named *imagesOnly* with **Initial Value** set to **true**, the toolbox page displays without the text labels next to the icons
- **isCommon**: if you give a toolbox page an attribute named *IsCommon* with **Initial Value** set to **true**, the toolbox page is common to all defined toolboxes while your technology is active; the page is initially displayed as collapsed
- **isCollapsed**: if you give a toolbox page an attribute named *IsCollapsed* with **Initial Value** set to **true**, the toolbox page is initially minimized.
- **Icon**: see [Assign Icons for Toolbox Items](#) ^[1342]
- **isHidden**: see [Create Hidden Sub-Menus](#) ^[1341].

Enterprise Architect enables you to add custom compartments to elements, via a Toolbox profile in an MDG Technology.

Guide:

The following topics provide a guide and some examples of how to add custom compartments to elements.

Topic	Description	See also
Develop script	<p>At step 5 of the <i>Create Toolbox Profile</i> procedure, add an attribute called _image.</p> <p>Click on the (...) button next to the Initial Value field to display the shape script editor.</p> <p>Replace <code>shape main</code> with <code>shape ChildElement</code>.</p> <p>You can keep <code>shape main</code> if you prefer, to adjust some properties of the main element (such as color); however, the main shape then requires a call to DrawNativeShape() in order to work correctly.</p> <p>At this point, you can use the HasProperty query method to</p>	<p>Create Toolbox Profiles ^[1337]</p> <p>Shape Editor ^[1359]</p> <p>Drawing Methods ^[1366]</p> <p>Query Methods ^[1373]</p>

Topic	Description	See also
	<p>search child elements for specific properties (such as stereotypes) to be displayed in compartments.</p> <p>Below are two examples of shape scripts that you might use.</p>	
<p>Example 1 - Without Adjusting the Parent Element</p>	<pre data-bbox="507 510 1145 1205"> // Add compartments for Child elements. shape ChildElement { // Check if a child element has the property stereotype, if so set our compartment name to Properties. if (HasProperty("stereotype", "property")) { SetCompartmentName ("Properties"); } // Check if our child element has a public scope and add the + symbol if so to the child compartment. if (HasProperty("scope", "public")) { AppendCompartmentText ("+"); } // Add the child elements name to the child compartment. AppendCompartmentText("#NAME#"); } </pre> <p data-bbox="459 1240 1153 1357">The shape script checks all child elements to see if they have a stereotype of <i>property</i>. If this stereotype is found, the SetCompartmentName function sets a compartment called <i>Properties</i>.</p> <p data-bbox="459 1377 1118 1435">The script then checks whether the child has a public scope and, if it does, appends the + symbol.</p> <p data-bbox="459 1456 1150 1514">Finally, the AppendCompartmentText function adds the child's name to the compartment.</p> <p data-bbox="459 1534 1142 1709">If a compartment has already been declared by SetCompartmentName, any additional children that fall under the same compartment are automatically added to it without having to declare a new compartment name (that is, all children with the stereotype <i>property</i> end up in the <i>Properties</i> compartment).</p>	<p data-bbox="1171 1178 1409 1205">Drawing Methods </p>
<p>Example 2 - Adjust the Color of the Parent Element and Add Child Compartments</p>	<pre data-bbox="507 1798 1137 2029"> // Shape main affects the parent shape main { // Set the color of the parent element to red setfillcolor(255, 0, 0); // draw the parents native shape drawnativeshape(); } </pre>	

Topic	Description	See also
	<pre data-bbox="507 331 1102 853"> // Shape ChildElement adds Child Compartments to the parent. shape ChildElement { if (HasProperty("stereotype", "part")) { SetCompartmentName ("Parts"); } elseif (HasProperty ("stereotype", "mystereotype")) { SetCompartmentName("My Stereotype"); } AppendCompartmentText ("#NAME#"); } </pre> <p data-bbox="459 887 1153 972">The <code>shape main</code> section sets the color of the main element to red and then proceeds to add the child compartments as for Example 1.</p> <p data-bbox="459 992 1142 1137">In this case, however, the script checks whether a child has either the <i>Part</i> stereotype or the custom stereotype <i>MyStereotype</i> applied to it. If there are two child elements, one a <i>Part</i> and the other using <i>MyStereotype</i>, two compartments are created, called <i>Parts</i> and <i>My Stereotype</i>.</p> <p data-bbox="459 1158 932 1216">In order to display the compartments, AppendCompartmentText must be called.</p> <p data-bbox="459 1236 1137 1321">Note that SetCompartmentName and AppendCompartmentText do not accept new line characters; that is, the following call is not supported:</p> <pre data-bbox="507 1355 1158 1413"> SetCompartmentName(_T(" My Name \n My New Name")) </pre>	<p data-bbox="1169 1003 1410 1032">Drawing Methods^[1366]</p>
<p data-bbox="260 1469 435 1527">Complete the Toolbox Profile</p>	<p data-bbox="459 1469 1134 1527">After adding the shape script, continue following the steps for creating a Toolbox Profile and save and import the profile.</p> <p data-bbox="459 1547 1107 1632">You can then apply the shape script by simply dragging the element from the Toolbox onto the diagram and adding the appropriate children to that element.</p>	<p data-bbox="1169 1469 1345 1527">Create Toolbox Profiles^[1337]</p> <p data-bbox="1169 1547 1345 1606">Deploy An MDG Technology^[1357]</p>

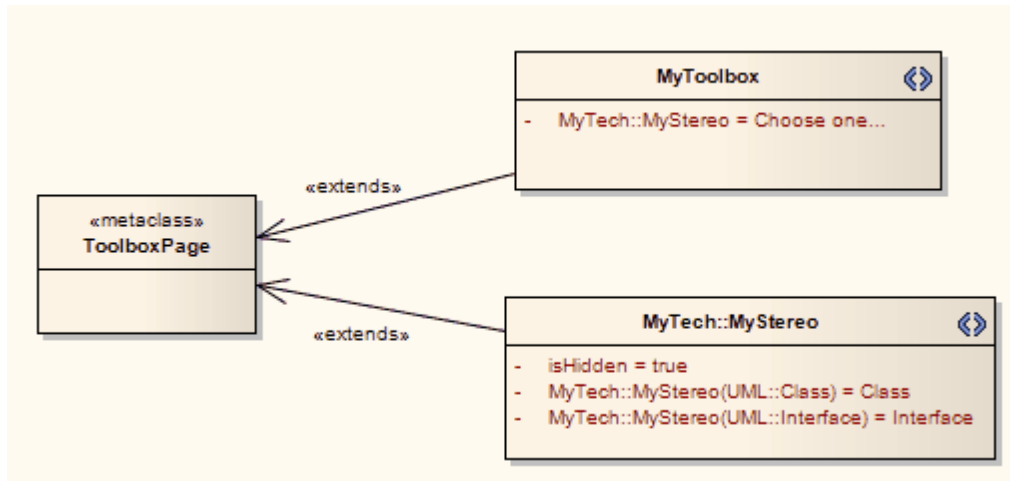
6.10.1.2.3.2 Create Hidden Sub-Menus

To create a sub-menu, create an additional «*stereotype*» element in the same toolbox profile and give it an attribute named *isHidden* with **Initial Value** of **true**. Define the toolbox item attributes as before.

In the parent «*stereotype*» element, create an attribute with the identical name to the sub-menu element. The sub-menu element can have an alias.

This technique is very useful for 'disambiguating' stereotypes that can be applied to multiple metaclasses. In the example below, the «*MyStereotype*» stereotype can be applied to either a Class or an Interface. On dragging and dropping one from the toolbox, a hidden menu displays giving the choice of Class or Interface, then the

appropriate element is dropped:



6.10.1.2.3.3 Override Default Toolboxes

Enterprise Architect has many default Toolbox Profiles, one for each of its inbuilt diagram types. These define the Toolbox pages that are displayed, by default, every time a diagram of a specific type is opened or brought into view.

To replace one of Enterprise Architect's default Toolboxes with one of your own (for example, if you have your own version of the UML::Class Toolbox that you want to be opened every time a Class diagram is opened - as long as your technology is active) then include a *RedefinedToolbox* clause in the **Notes** field for the diagram properties of your Toolbox Profile diagram.

For example, the profile diagram's **Notes** field could resemble the following:

```
RedefinedToolbox=UML::Class; Alias=Class; Notes=Structural elements for Class diagrams;
```

This states that the Toolbox defined by this profile replaces the Enterprise Architect Toolbox *UML::Class* as the default Toolbox for all UML Class diagrams. For a list of inbuilt Toolboxes, see the [List of Enterprise Architect Toolboxes](#)^[1343] topic.

6.10.1.2.3.4 Assign Icons To Toolbox Items

How to

To assign an icon to a toolbox item

Step	Action	See also
1	Create a new <i>«stereotype»</i> element in the same toolbox profile as the toolbox item.	
2	Give the <i>«stereotype»</i> element the same name as the attribute that it is assigning an image to; for example, in the diagram below:	

Step	Action	See also
	MyTech::MyStereo(UML::Class)	
3	Give the «stereotype» element an attribute named <i>Icon</i> with Initial Value set to the full path and file name of the image to be used. The icon image is a 16x16 pixel bitmap file; for a transparent background, use light grey - RGB(192,192,192).	
4	Create an Extends association from the «stereotype» element to a «metaclass» element named <i>ToolboxItemImage</i> .	



6.10.1.2.3.5 List of Enterprise Architect Toolboxes

The following is a list of the Toolbox pages that can be overridden:

- UML::Activity
- UML::Class
- UML::Communication
- UML::Component
- UML::Composite
- UML::Deployment
- UML::Interaction
- UML::Metamodel
- UML::Object
- UML::Profile
- UML::State
- UML::Timing
- UML::UseCase
- Extended::Analysis
- Extended::Custom
- Extended::DataModeling
- Extended::Maintenance
- Extended::Requirements
- Extended::UserInterface
- Extended::WSDL
- Extended::XMLSchema

6.10.1.2.3.6 Elements Used in Toolboxes

The following elements (all preceded with the namespace **UML::**) can be extended or redefined in Enterprise Architect Toolbox pages. The text in red indicates the label name displayed in the default Enterprise Architect Toolbox pages, where this differs in any way from the **UML::** statement text.

When these profile elements are incorporated into an MDG Toolbox page, they enable [synchronization](#)^[1287] of the Tagged Values and Constraints of all elements created from them.

You can also extend [connectors](#)^[1345].

- Action
- Activity
- ActivityFinal (**Final**)
- ActivityInitial (**Initial**)
- ActivityParameter
- ActivityPartition (**Partition**)
- ActivityRegion (**Region**)
- Actor
- Artifact
- AssociationElement (**Association**)
- Boundary (for Use Cases)
- CentralBufferNode (**Central Buffer Node**)
- Change
- Choice
- Class
- Collaboration
- CollaborationOccurrence (**Collaboration Use**)
- Comment (**Note**)
- Component
- Constraint
- Datastore
- Decision
- DeploymentSpecification (**Deployment Specification**)
- Device
- DiagramLegend (**Diagram Legend**)
- DiagramNotes (**Diagram Notes**)
- DocumentArtifact (**Document Artifact** or **Document**)
- Entity (**Information**)
- EntityObject (**Entity**)
- EntryState (**Entry**)
- InteractionState (**State/Continuation**)
- Interface
- Issue
- Junction
- Lifeline
- MergeNode (**Merge**)
- MessageEndPoint (**Endpoint** or **Message Endpoint**)
- MessageLabel (**Message Label**)
- Metaclass
- Node
- Object
- ObjectBoundary (**Boundary**)
- ObjectControl (**Control**)
- ObjectEntity (**Entity**)
- Package
- PackagingComponent
- Part
- Port
- Primitive
- Process
- Profile
- ProvidedInterface (**Expose Interface**)
- ReceiveEvent (**Receive**)
- Requirement
- RobustBoundary (**Boundary**)
- RobustControl (**Control**)
- RobustEntity (**Entity**)
- Screen
- SendEvent (**Send**)
- SequenceBoundary (**Boundary**)
- SequenceControl (**Control**)

- Enumeration
- ExceptionHandler (**Exception**)
- ExecutionEnvironment (**Execution Environment**)
- ExitState (**Exit**)
- Feature
- FinalState (**Final**)
- FlowFinalNode (**Flow Final**)
- ForkJoinH (**Fork/Join** - Horizontal)
- ForkJoinV (**Fork/Join** - Vertical)
- Gate (**Diagram Gate**)
- GUIElement (**UI Control**)
- HistoryState (**History**)
- Hyperlink
- InformationItem (**Information Item**)
- InitialState (**Initial**)
- InteractionFragment (**Fragment**)
- SequenceEntity (**Entity**)
- Signal
- State
- StateMachine (**State Machine**)
- StateTimeLine (**State Lifeline**)
- Stereotype
- StructuredActivity (**Structured Activity**)
- SynchState (**Synch**)
- Table
- Terminate
- TestCase (**Test Case**)
- Text
- UseCase (**Use Case**)
- UMLBoundary (**Boundary**)
- ValueTimeLine (**Value Lifeline**)

6.10.1.2.3.7 Connectors Used In Toolboxes

The following connectors (all preceded with the namespace **UML::**) can be extended or redefined in Enterprise Architect toolboxes. The text in red indicates the label name displayed in the default Enterprise Architect Toolbox pages, where this differs in any way from the **UML::** statement text.

You can also extend [elements](#) ^[1344].

- Aggregation (**Aggregate**)
- Assembly
- Association (**Associate**)
- AssociationClass (**Association Class**)
- CallFromRecursion (**Call**)
- CommunicationPath (**Communication Path**)
- Composition (**Compose**)
- Connector
- ControlFlow (**Control Flow**)
- Delegate
- Dependency
- Deployment
- Extension
- Generalization (**Generalize** or **Inheritance**)
- InformationFlow (**Information Flow**)
- InterruptFlow (**Interrupt Flow**)
- Invokes
- ObjectFlow (**Object Flow**)
- Occurrence
- PackageImport (**Package Import**)
- PackageMerge (**Package Merge**)
- Precedes
- ProfileApplication (**Application**)
- Realization (**Realize** or **Implements**)
- Recursion
- Redefinition
- Representation
- Represents
- RoleBinding (**Role Binding**)
- SelfMessage (**Self-Message**)
- TagValAssociation (**Tagged Value**)
- TemplateBinding (**Template Binding**)
- TraceLink (**Trace**)
- Transition

- Manifest
- Message
- Nesting
- NoteLink (**Note Link**)
- UCExtend (**Extend**)
- UCInclude (**Include**)
- UseCaseLink (**Use**)

6.10.1.2.4 Create Custom Diagram Profiles

You can create extended diagram types in Enterprise Architect and include them in MDG Technologies.

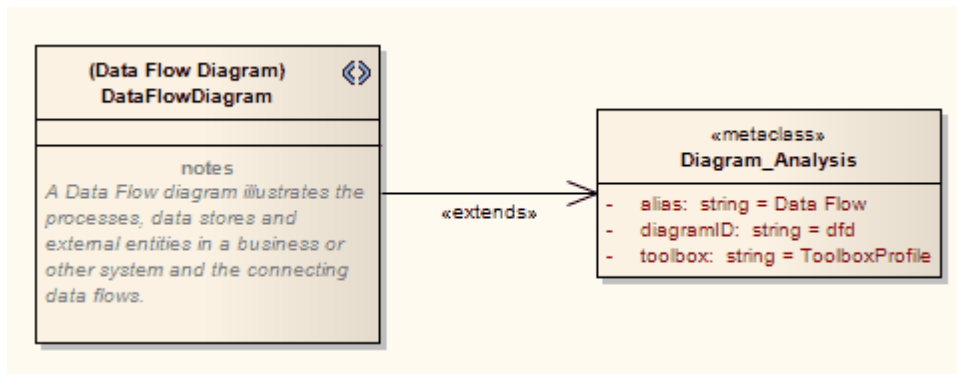
How to

To create extended diagram types

Step	Action	See also
1	Create a profile with the same name as the MDG Technology in which it is to be included; for example, <i>SysML</i> . However, check the Note in the <i>Defining Child Diagram Types</i> topic.	Defining Child Diagram Types ^[1315]
2	Create a « <i>stereotype</i> » Class element that is named as the custom diagram, for example, <i>BlockDefinition</i> .	
3	Create a Class element and name it as one of the Built-In Diagram Types prefixed with <i>Diagram_</i> . For example <i>Diagram_Logical</i> for Class diagrams or <i>Diagram_Use Case</i> for Use Case diagrams.	Built-In Diagram Types ^[1348]
4	Give the <i>Diagram_x</i> Class the « <i>metaclass</i> » stereotype and draw an « <i>extends</i> » connector from the stereotype to the metaclass.	
5	In the Notes field, give the stereotype Class a brief description of what the diagram is used for. This description displays in the bottom right-hand corner of the New Diagram dialog.	
6	Give the <i>Diagram_x</i> Class the following attributes as required: <ul style="list-style-type: none"> • <i>alias: string = Type</i> (where <i>Type</i> appears before the word 'Diagram' on the diagram title bar) • <i>diagramID: string = abc</i> (where <i>abc</i> is the diagram type that appears in the diagram frame label) • <i>toolbox: string = ToolboxName</i> (where <i>ToolboxName</i> is the name of the toolbox profile for the toolbox that opens automatically each time a diagram is opened) • <i>frameString: string = FrameFormatString</i> (where 	Diagram Frame ^[1119] Attribute Values - StyleEx & pdata ^[1348]

Step	Action	See also
	<p><i>FrameFormatString</i> is a string containing substitution macros for defining the frame title, with or without additional delimiters such as (); macros that can be used are:</p> <ul style="list-style-type: none"> • #DGMSTEREO# • #DGMD# • #DGMTYPE# • #DGMALIAS# • #DGMOWNERNAME# • #DGMOWNERNAMEFULL# • #DGMNAME# • #DGMNAMEFULL# <ul style="list-style-type: none"> • <i>swimlanes</i>: <i>string</i> = <i>Lanes=2;Orientation=Horizontal; Lane1=Title1;Lane2=Title2</i>; (where <i>Lanes</i> can be any value, but the number of <i>Lane<n></i> values must equal the value of <i>Lanes</i>; <i>Orientation</i> can be omitted, in which case the swimlanes default to vertical) • <i>StyleEx</i>: <i>string</i> = one or more of a range of values; see the <i>Attribute Values - StyleEx & pdata</i> topic • <i>pdata</i>: <i>string</i> = one or more of a range of values; see the <i>Attribute Values - StyleEx & pdata</i> topic 	
7	Save the diagram as a profile in the usual manner.	
8	Add the diagram profile to the .MTS file used in the MDG Technology.	Add Diagram Types <small>[1329]</small>

The following example shows the DFD diagram profile which defines a DFD diagram as an extension of the Enterprise Architect Analysis diagram.



6.10.1.2.4.1 Built-In Diagram Types

The following is a full list of built-in diagram types provided by Enterprise Architect.

- Activity
- Analysis
- Collaboration
- Component
- CompositeStructure
- Custom
- Deployment
- InteractionOverview
- Logical
- Object
- Package
- Sequence
- Statechart
- Timing
- Use Case

Note the use of *Logical* for Class diagrams and also notice the space in the middle of *Use Case*. These names are used in [Defining Child Diagram Types](#)^[1315], or prefixed by *Diagram_* in creating [Diagram Profiles](#)^[1346].

6.10.1.2.4.2 Attribute Values - StyleEx & pdata

When creating a diagram profile, you can use the *pdata* and *StyleEx* attributes to define a range of characteristics of the diagrams created with the profile. If the attribute is defining several characteristics at once, put the values in a single string separated by semicolons.

StyleEx: string = HideQuals=0;AdvanceElementProps=1;ShowNotes=1;

StyleEx: string =

- *AdvancedConnectorProps=1;* (to show connector property strings)

- *AdvancedElementProps=1*; (to show the element property string)
- *AdvancedFeatureProps=1*; (to show the feature property string)
- *AttPkg=1*; (to show package visible Class members)
- *DefaultLang=Language*; (to set the default language for the diagram; *Language* can be one of Enterprise Architect's built-in languages such as C++ or Java, or it can be a custom language)
- *HandDrawn=1*; (to apply hand drawn mode)
- *HideQuals=0*; (to show qualifiers and visibility indicators)
- *SeqTopMargin=50*; (to set the height of the top margin on sequence diagrams)
- *ShowAsList=1*; (to make the diagram open directly into the [Diagram List](#)⁽⁵⁹⁶⁾)
- *ShowMaint=1*; (to show the element Maintenance compartment)
- *ShowNotes=1*; (to show the element Notes compartment)
- *ShowOpRetType=1*; (to show the operation return type)
- *ShowTests=1*; (to show the element Testing compartment)
- *SuppConnectorLabels=1*; (to suppress all connector labels)
- *SuppressBrackets=1*; (to suppress brackets on operations without parameters)
- *TConnectorNotation=Option*; (where *Option* is one of **UML 2.1**, **IDEF1X**, or **Information Engineering**)
- *TExplicitNavigability=1*; (to show non-navigable connector ends)
- *VisibleAttributeDetail=1*; (to show attribute details on the diagram)
- *Whiteboard=1*; (to apply whiteboard mode)

pdata: string =

- *HideAtts=0*; (to show the element Attributes compartment)
- *HideEStereo=0*; (to show element stereotypes in the diagram)
- *HideOps=0*; (to show the element Operations compartment)
- *HideParents=0*; (to show additional parents of elements in the diagram)
- *HideProps=0*; (to show property methods)
- *HideRel=0*; (to show relationships)
- *HideStereo=0*; (to show attribute and operation stereotypes)
- *OpParams=3*; (to show operation parameters)
- *ShowCons=1*; (to show the element Constraints compartment)
- *ShowIcons=1*; (to use stereotype icons)
- *ShowReqs=1*; (to show the element Requirements compartment)
- *ShowSN=1*; (to show sequence notes)
- *ShowTags=1*; (to show the element Tagged values compartment)
- *SuppCN=0*; (to show collaboration numbers)
- *UseAlias=1*; (to use the aliases or elements in the diagram, if available)

6.10.1.2.5 Create Learning Center Profiles

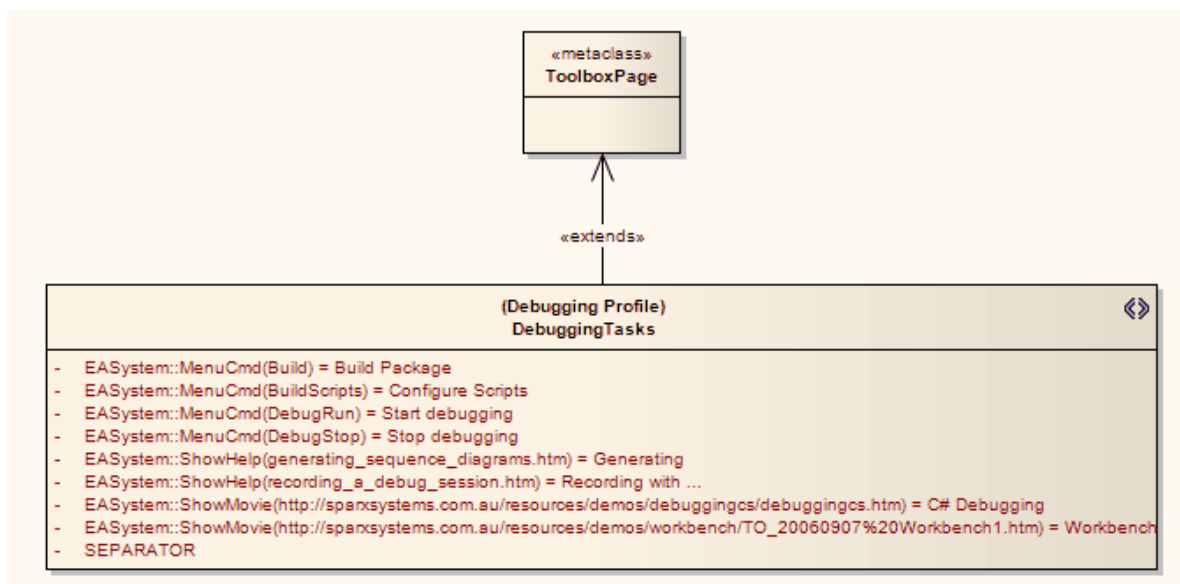
Defining Learning Center profiles is a two-part process:

Step	Action	See also
1	Define toolboxes; create any number of stereotype elements that each define a Learning Center Toolbox page.	Define Learning Center Toolboxes ^[1350]
2	Create the profile and incorporate it into your Technology.	Save Learning Center Profile ^[1354] Add Learning Center Pages ^[1330]

6.10.1.2.5.1 Define Learning Center Toolboxes

A Learning Center toolbox is defined by a «*stereotype*» Class that extends a «*metaclass*» ToolboxPage element. These elements must be owned by a «*profile*» package.

Each «*stereotype*» Class represents the contents of a Learning Center topic and each attribute of the «*stereotype*» Class defines a command button in the Learning Center. The following diagram shows an example of a Learning Center toolbox.



The title bar of the Learning Center toolbox is defined by the Alias of the «*stereotype*» Class; in the above diagram, *Debugging Profile*.

The content of the Learning Center page is defined by the following standard attribute types:

- **EASystem::ShowDoc(folder\filename.rtf)** - these entries provide access to RTF documents (or any

other text/document files), where *folder* is relative to the install folder and *filename* is the name of the (RTF) text file; this is a very significant attribute in creating Learning Center topics

- **EASystem::MenuCmd**. - these entries name an Enterprise Architect main menu command inside round brackets - see the complete list of built-in Learning Center commands; type the text to appear in the Learning Center into the **Initial Value** field
- **EASystem::ShowHelp** - these entries name a page from the *Enterprise Architect User Guide* inside round brackets - to find out the names of pages in the *Enterprise Architect User Guide*, right-click on the page and select the **Properties** context menu option; type the text to appear in the Learning Center into the **Initial Value** field
- **EASystem::ShowMovie** - these entries give the URL of a movie inside round brackets - type the text to appear in the Learning Center into the **Initial Value** field
- **EASystem::ShowView(command)** - these entries open an Enterprise Architect window, where *command* is one of the View commands from the second list of built-in Learning Center commands
- **EASystem::ShowURL** - this entry gives the URL of a web page inside round brackets; type the text to appear in the Learning Center into the **Initial Value** field
- **isCommon**: - a boolean attribute with **Initial Value** set to **True**, defines a Learning Center toolbox as common
- **SEPARATOR** - this entry indicates that a separator should be placed in the Learning Center toolbox; if it is necessary to place multiple separators in a single toolbox, note that Enterprise Architect does not allow identically named attributes for a Class: simply change the case of one or more letters to get around the problem

You can also run Add-In functions from the Learning Center.

Next Step:

The next step is to save the Learning Center Profile.

Learn more

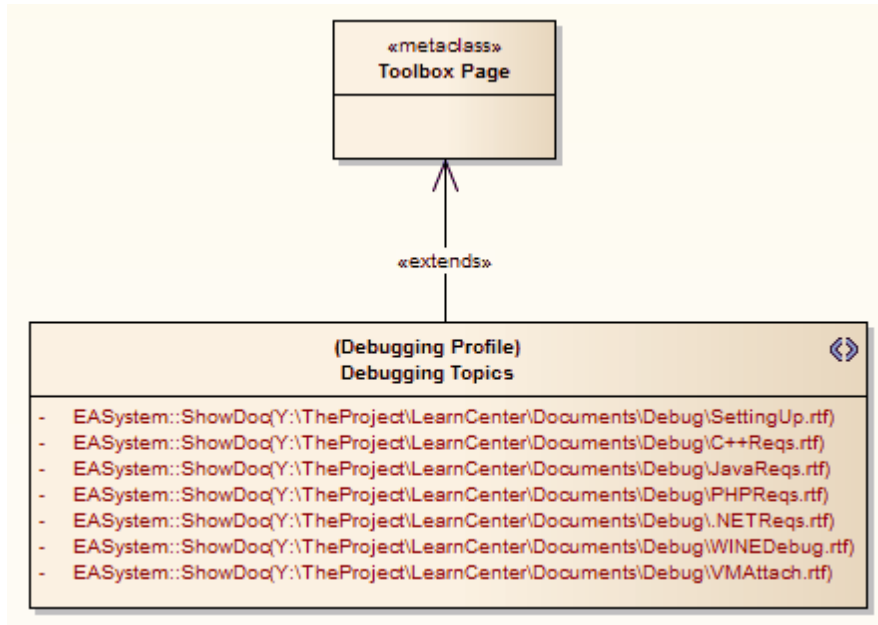
- [Define Links To Documents](#) ^[1351]
- [Built-In Learning Center Commands](#) ^[1352]
- [Run Add-In Functions](#) ^[1353]
- [Save a Learning Center Profile](#) ^[1354]

The primary purpose of the Learning Center is to provide information on specific topics, processes and actions, and the easiest way to present this information is in the form of one or more structured documents.

Therefore, within your Learning Center profile, you would probably have «*stereotype*» Classes containing several instances of the *ShowDoc* attribute; that is:

- *EASystem::ShowDoc(folderfilename.rtf)*

These attributes identify buttons that provide access to text or document files, where *folder* is relative to the install folder and *filename* is the name of the (RTF) text file containing your structured document.



The following Enterprise Architect commands can all be used in user-defined Learning Center profiles.

Learning Center pages that call functions have attributes named in the form *EASystem::MenuCmd* (<CommandName>) where <CommandName> is the name chosen from the following list:

- AddDiagram
- AddElement
- AddPackage
- AutoRecordThread
- AddModelFromPattern
- Build
- BuildScripts
- ConfigureCSV
- ConfigureValidation
- CreateBaseLine
- CreateSequenceDiagram
- DebugPause
- DebugRun
- DebugStop
- Deploy
- DiagramsOnlyReport
- ElementUsage
- ExportXML
- FileNew
- FileOpen
- GenerateDDL
- GenerateWSDL
- GenerateXMLSchema
- ImplementationDetails
- ImportBinary
- ImportExportCSV
- ImportSchema
- ImportSourceDirectory
- ImportWSDL
- ImportXML
- ImportXMLSchema
- Run
- RunHTMLReport
- RunRTFReport
- SetClassifier
- ShowHideExecution
- StartDebugRecording
- StepInto
- StepOut
- StepOver
- StopDebugRecording
- Test
- TestingReport
- ToggleLevelNumbering
- TransformPackage
- TransformSelectedElements
- ValidateModel

Learning Center pages that call Enterprise Architect windows have attributes named in the form *EASystem:: ShowView(<Command>)* where *<Command>* is the name chosen from the following list:

- ViewAuditing
- ViewBreakpoints
- ViewCallstack
- ViewDebug
- ViewDiagramAsList
- ViewDiagramLayout
- ViewDiagramToolBox
- ViewElementList
- ViewFileSearch
- ViewForum
- ViewGapAnalysis
- ViewHierarchy
- ViewLocals
- ViewMaintenance
- ViewMemory
- ViewModelViews
- ViewModules
- ViewNotes
- ViewOutput
- ViewPackageScripts
- ViewPanAndZoom
- ViewProfiler
- ViewProjectBrowser
- ViewProjectCalendar
- ViewProjectInfo
- ViewProjectManagement
- ViewPropertiesWindow
- ViewRecording
- ViewRelationships
- ViewRelMatrix
- ViewRequirementTypes
- ViewResources
- ViewRules
- ViewScripting
- ViewSearch
- ViewSourceCode
- ViewTaggedValues
- ViewTaskAllocations
- ViewTesting
- ViewTestingDetails
- ViewTestpoints
- ViewTransformationTemplates
- ViewWatches
- ViewWebBrowser
- ViewWorkbench

To run Add-In functions from the Learning Center, you create an attribute in the Learning Center «*stereotype*» Class with the following format:

```
" Assembly : : FunctionName ( ) "
```

where *Assembly* is the name of the Add-In and *FunctionName* is the name of a public function in the Add-In. Give the attribute an initial value of the text that is to appear in the Learning Center.

The function receives two parameters and returns a success status, as in the following VB.Net example:

```
Public Function ShowMyDiagram(ByRef Repository As EA.Repository, ByVal
args As Object) As String
    Dim ret As String
    ret = Repository.SQLQuery("select ea_guid from t_diagram where
diagram_type='Custom' and StyleEx like
'*;MDGDgm=MyDiagrams::MyCustomDiagram;*'")
    If ret Is Nothing Then
        ShowMyDiagram = False
        Exit Function
    End If

    Dim oXML As MSXML2.DOMDocument = New MSXML2.DOMDocument
    oXML.LoadXML(ret)

    Dim NodeList As MSXML2.IXMLDOMNodeList = oXML.selectNodes("//
ea_guid")
    If NodeList.Length = 0 Then
        ShowMyDiagram = False
        Exit Function
    End If

    Dim Node As MSXML2.IXMLDOMNode
    Dim diag As EA.Diagram
    If NodeList.Length >= 1 Then
        Node = NodeList.item(0)
```

```

    di ag = Reposi tory. Get Di agr amByGui d( Node. t ext )
    Reposi tory. OpenDi agr am( di ag. Di agr amI D)
    Reposi tory. ShowI nPr oj ect Vi ew( di ag)
  End I f

  ShowMyDi agr am = Tr ue
End Functi on

```

6.10.1.2.5.2 Save a Learning Center Profile

The best organization structure for the model in which you are creating your Learning Center Profile is:

- A single «*profile*» package
- Three diagrams within the «*profile*» package named *Toolboxes*, *Contexts* and *Context Allocations*
- Each toolbox page «*stereotype*» element is owned by the «*profile*» package and appears on the *Toolboxes* diagram
- Each «*metaclass*» element is owned by the «*profile*» package and appears on the *Toolboxes* diagram.

From this structure, creating a Learning Center Profile is as simple as right-clicking on the «*profile*» package in the Project Browser and selecting the **Save Package as UML Profile** context menu option.

6.10.1.2.6 Define Validation Configuration

The Model Validation Configuration dialog can be opened using the **Project | Model Validation | Configure...** menu option.

Using this dialog, you can choose which sets of validation rules are and are not executed when a user performs a validation. Rather than perform this configuration manually and potentially have to change the settings every time Enterprise Architect is started and a different technology is set active, you can define the configuration settings within the MTS file.

White List:

To specify a set of rules as a white-list (that is, anything added to this list is turned ON), open your MTS file in a text editor and copy and paste the following `<Model Val i dat i on>` block at the top level inside the `<MDG. Sel ect i ons>` block:

```

<Model Val i dat i on>
  <Rul eSet name=" BPMNRul es" / > <!-- rul eset I D defined i n the Pr oj ect .
  Def i neRul eCat egor y call -->
  <Rul eSet name=" MVR7F0001" / > <!-- notice you can turn on/ off system
  rul es as well ! -->
</ Model Val i dat i on>

```

Ensure that the ruleset IDs do not contain any spaces.

Black List:

To specify a set of rules as a black-list (that is, anything added to this list is turned OFF), open your MTS file in a text editor and copy and paste the following `<Model Val i dat i on>` block at the top level inside the `<MDG. Sel ect i ons>` block:

```

<Model Val i dat i on i sBl ackLi st=" true" >
  <Rul eSet name=" BPMNRul es" / >
  <Rul eSet name=" MVR7F0001" / >

```

```
</ Model Val i dat i on>
```

In the examples above, "BPMNRu l es" is the rule-set ID defined in the Pr oj ect .
 Def i neRu l eCat egor y call - see [Project Interface](#)^[2535] for details. "MVR7F0001" is one of Enterprise Architect's built-in rule-sets. These validation options are applied when you activate the appropriate technology. The global (default) technology has all rules turned on.

6.10.1.2.7 Incorporate Model Templates

Model templates can be added into a model, either on creation of the model, or at any time by right-clicking on a package in the Project Browser and selecting the **Add | Add a New Model using Wizard...** context menu option.

To include your own model templates in this dialog, add them to your MDG Technology as described in the following table:

Step	Action	See also
1	<p>Create a package that contains all sub-packages, diagrams, elements, notes and information links that you want to provide in the model template.</p> <p>See the <i>EAExample.eap</i> model for illustrations of what you might include, or create a model from a standard template and see what is generated.</p> <p>Your model template package must be self contained and not contain any dependencies or other links to elements outside the package.</p>	Model Wizard ^[660]
2	Export your package to XML.	Export to XML ^[413]
3	<p>Create a reference to the XML file in the MTS file; open your MTS file in a text editor and copy and paste the following <code><Model Templates></code> block at the top level inside the <code><MDG. Sel ect i on s></code> block:</p> <pre><Model Templates> <Model name=" Template Name" description=" This is the description. " location=" MyTemplatePackage.xml " default=" yes" icon = " 34" filter= " Filter Name" /> </ Model Templates></pre>	Working with MTS Files ^[1335]

You can include as many `<Model Templates>` blocks in your MTS file as you have model templates. The attributes have the following meanings:

- **Model name:** The name of the model template to show in the Model Wizard, which displays when you create a new model or when you execute the **Add a New Model using Wizard** menu option
- **description:** The text to display in the Model Wizard when the name is selected
- **location:** The path of the XML file that contains the XML export of the model template package, relative to the location of the MDG Technology file; if the XML file is in the same folder as the technology file then this just contains the file name

- **default:** Contains either **yes** to indicate that the model template is checked by default, or **no** to indicate that the model template is un-checked by default
- **icon:** Contains an index to Enterprise Architect's base icons list; to show the appropriate view icon, use one of the following values: **29** = Use Case, **30** = Dynamic; **31** = Class; **32** = Component; **33** = Deployment; **34** = Simple
- **filter:** If you have a large number of model templates, you can group them on the Model Wizard by giving all the model templates in the same group the same filter name; the filter name given appears in the **Select from:** list box in the Model Wizard

6.10.1.2.8 Set Up Technology Element Images

This topic outlines how to capture images to represent MDG Technology elements, so that the images are displayed on a diagram a user creates through the technology when it is deployed in the user's model. Note that you would probably work backwards and forwards through the steps many times, adding objects as you identify the requirement for them.

To capture images to represent MDG Technology elements

Step	Action	See also
1	Import suitable images into the MDG Technology development model from various sources, using the Add New button on the Image Manager.	Image Manager ^[755]
2	Design and create a Stereotype (UML) Profile containing (if appropriate) a stereotype definition for each element or connector to be owned by the technology. These stereotype definitions can contain Shape Scripts that in turn incorporate the imported images.	Stereotype (UML) Profile ^[1293] Shape Scripts ^[1304] Drawing Methods ^[1366]
3	Design and create a Toolbox Profile with stereotype elements that contain an attribute for each element or connector that can be dropped onto a diagram from the toolbox. These attributes identify the name of the technology element or connector, any modifying stereotype (which might incorporate the required image) and the UML or Extended element or connector on which the technology object is based. For example: <i>SysML::Block(UML::Class)</i> , where <i>SysML</i> is the Technology Profile, <i>UML::Class</i> is the UML element used as the base, and <i>Block</i> is the stereotype that modifies the Class to turn it into a SysML Block element.	Toolbox Profiles ^[1337]
4	Design and create a Diagram Profile that identifies the Toolbox Profile. When a diagram of the type defined in the Diagram Profile is opened, it in turn opens a set of toolbox pages as defined by the Toolbox Profile.	Diagram Profiles ^[1346]
5	Create or update the technology as required, adding the Stereotype Profile, Diagram Profile, Toolbox Profile and Image files to the technology from the development model.	Creating MDG Technologies ^[1324] Adding Stereotype Profile ^[1327]

Step	Action	See also
		Adding Diagram Profile ^[1329] Adding Toolbox Profile ^[1329] Adding Image Files ^[1333]
6	<p>Deploy the technology as appropriate.</p> <p>When a user applies the technology to their own model, and creates a diagram under that technology, the elements they create on the diagram should be represented by the images you assigned to those elements when you created the technology.</p>	Deploying MDG Technologies ^[1357]

6.10.1.2.9 Deploy An MDG Technology

An MDG Technology can be deployed in one of two ways: as a file or from an Add-In.

Deploy From a File:

To deploy your technology as a file, you have a number of choices:

- Import it into the %APPDATA% folder; see the [Import MDG Technologies](#) ^[1288] topic
- Copy it to a folder named MDGTechnologies, which you must create under your Enterprise Architect installation directory (by default this is C:\Program Files\Sparx Systems\EA; when you restart Enterprise Architect, your MDG Technology is deployed)
- Copy it to any folder in your file system, including network drives - use the Enterprise Architect **Settings | MDG Technologies...** menu option, press the **Advanced** button and add the folder to the Technologies path; this deployment method enables you to quickly and easily deploy a technology to all Enterprise Architect users on a LAN
- Upload it to an internet or intranet location: use the Enterprise Architect **Settings | MDG Technologies...** menu option, press the **Advanced** button and add the URL to the Technologies path; this deployment method enables you to quickly and easily deploy a technology to an even wider group of Enterprise Architect users

Deploy From an Add-in:

To deploy your technology from an Add-In, you must write an [EA_OnInitializeTechnologies](#) ^[2663] function. The following example is written in VB.Net:

```
Public Function EA_OnInitializeTechnologies(ByVal Repository As EA.
Repository) As Object
    EA_OnInitializeTechnologies = My.Resources.MyTechnology
End Function
```

6.10.1.3 Shape Scripts

Introduction

Enterprise Architect *Shape Scripts* enable you to specify custom shapes via a scripting language. These custom shapes are drawn instead of the standard UML notation. Each script is associated with a particular stereotype, and is drawn for every element of that stereotype.

The following topics describe how to create and apply Shape Scripts:

- [Getting Started with Shape Scripts](#) ^[1358]
- [Shape Editor](#) ^[1359]
- [Shape Script Overview](#) ^[1360]
- [Write Scripts](#) ^[1363]
- [Example Scripts](#) ^[1380]
- [Add Shape Scripts to UML Profiles](#) ^[1304]

Learning Center topics

- (Alt+F1) | [Modeling Languages | Defining Scripts](#)

6.10.1.3.1 Getting Started With Shape Scripts

Shape Scripts are associated with stereotypes and are defined via the Stereotypes tab of the UML Types dialog. Each stereotype defined can have a Shape Script.

Access [Settings | UML Types > Stereotypes: Shape Script](#)

Step	Action	See also
1	You can create a Shape Script for an existing stereotype by selecting the stereotype from the list. Alternatively, you can create new stereotypes by clicking on the New button and giving the stereotype a name - select a base Class and click on the Save button. Once the stereotype is saved, it displays in the list.	
2	To override the appearance, select the Shape Script radio button and then click on the Assign button. The Shape Script Editor displays.	Shape Script Overview ^[1360] Shape Script Editor ^[1359]
3	Type the Shape Scripts in the Edit window. Click on the Refresh button in order to view the shape in the preview window.	Writing Scripts ^[1363]
4	If you define a composite Shape Script (such as the connector at the end of the <i>Example Scripts</i> topic), click on the Next Shape button to page through the components of the shape.	Example Scripts ^[1380]

Step	Action	See also
5	Once you have finished writing your Shape Script, click on the OK button. To save the Shape Script you must click on the Save button on the Stereotypes tab.	
6	Once you have created your Shape Script for a particular stereotype, you can assign that stereotype to an element or connector; the appearance reflects the Shape Script you created. To do this, drag and drop the appropriate element or connector into your diagram.	
7	Right-click on the element or connector and select the Properties context menu option. Click on the Stereotype field drop-down arrow, select the stereotype you created and click on the OK button. The object's shape now reflects the Shape Script you created.	

Notes

- If an element's appearance is modified by a Shape Script, many of the options on the **Advanced** context menu for that element are disabled

6.10.1.3.2 Shape Editor

The Shape Editor enables you to create Shape Scripts. It provides the facilities of the *Common Code Editor*, including intellisense for Shape Script attributes and functions.

You create the Shape Script on a stereotype that you type in or select on the **Stereotype** field on the UML Types dialog.

Access [Settings | UML Types > Stereotypes \(specify stereotype\): Shape Script, Assign](#)

Reference

Field	Usage	See also
Format	Select the Shape Script version.	
Import	Import a Shape Script from a text file.	
Export	Export a Shape Script to a text file.	

Field	Usage	See also
OK	Exit from the Shape Editor, don't forget to save your script on the Stereotypes tab.	Getting Started With Shape Scripts ^[1358]
Next Shape	Rotate though the multiple shape definitions.	
Refresh	Parse your script and display the result in the Preview window.	

Learn more

- [Editing Source Code](#) ^[1799], for more information on intellisense and the Common Code Editor

6.10.1.3.3 Shape Script Overview

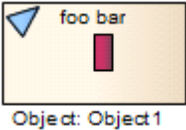
Using Shape Scripts you can define a model image for elements and connectors.

Topics

Topic	Description	See also
Abstract	<p>Shape Scripts contain a number of sections for defining different aspects of the image produced; for an element these include:</p> <ul style="list-style-type: none"> • Main object • Label • Decoration (for example, a Document element might contain an icon depicting a document) <p>For a connector these include:</p> <ul style="list-style-type: none"> • Main object • Shape Source • Shape Target • Labels <p>Shape Scripts work on the basis that a default pattern is maintained unless a script contains an alternative definition</p> <ul style="list-style-type: none"> • If you have a Shape Script with just a decoration, this decoration is added on top of the normally-drawn object • If you have an empty shape routine, it overrides the default; so, a blank 'shape label' prevents the creation of the floating text label for elements that have them 	

Topic	Description	See also
	Scripting is NOT case-sensitive: 'Shape' is the same as 'shape'.	
Block Layout: Element Example	<p>Below is a simple block example of a script for an element:</p> <pre data-bbox="486 481 1212 869"> shape main { // draw the object } shape label { // draw a floating text label } decoration <identifier> { // draw a 16x16 decoration inside the object } </pre> <p>The <code><identifier></code> string is an alphanumeric word.</p>	
Block Layout: Connector Example	<p>For a connector the example is:</p> <pre data-bbox="486 1093 1252 1585"> shape main { // draw the line } shape target { // draw the shape at the target end } shape source { // draw the shape at the source end } label <positionLabel> { // define the text for the label } </pre> <p>The <code><positionLabel></code> string can be any of:</p> <ul data-bbox="486 1686 734 1955" style="list-style-type: none"> • lefttoplabel • leftbottomlabel • middletoplabel • middlebottomlabel • righttoplabel • rightbottomlabel 	

Topic	Description	See also
Order of declaration	Shapes can consist of Attribute declarations, Command calls and Sub Shape definitions, which must appear in that order; that is, Attribute declarations must appear before all Command calls and Sub Shape definitions must appear last.	Shape Attributes Sub-Shapes
Sub Shapes	<p>A shape can have Sub Shapes.</p> <p>Sub Shapes must be declared after the main Shape Script, but called from the Method commands.</p> <p>The following is an example of the ordering for declarations:</p> <pre> shape main { //Initialisation Attributes - these must be before drawing commands noshadow = "true"; h_align = "center"; //drawing commands (Methods) rectangle(0,0,100,100); println("foo bar"); // call the sub-shape addsubshape("red", 20, 70); // definition of a sub-shape shape red { setfillcolor(200, 50, 100); rectangle(50,50,100,100); } //definition of a label shape label { setOrigin("SW", 0, 0); println("Object: #NAME#"); } //definition of a Decoration decoration triangle { // Draw a triangle for the decoration startpath(); moveto(0,30); lineto(50,100); lineto(100,0); endpath(); setfillcolor(153,204,255); fillandstrokepath(); } </pre>	Drawing Methods

Topic	Description	See also
		

Learn more

- [Write Scripts](#) ^[1363]
- [Example Scripts](#) ^[1380]

6.10.1.3.4 Write Scripts

This topic is a detailed reference for writing Shape Scripts. For an introduction to writing Shape Scripts, see the *Getting Started With Shape Scripts* and *Example Scripts* topics.

See the following reference topics for more detailed information on shape scripting:

- [Shape Attributes](#) ^[1363]
- [Drawing Methods](#) ^[1366]
- [Color Queries](#) ^[1372]
- [Conditional Branching](#) ^[1373]
- [Query Methods](#) ^[1373]
- [Display Element/Connector Properties](#) ^[1373]
- [Sub-shapes](#) ^[1376]
- [Reserved Names](#) ^[1377]
- [Miscellaneous](#) ^[1378]
- [Syntax Grammar](#) ^[1379]

Learn more

- [Getting Started With Shape Scripts](#) ^[1358]
- [Example Scripts](#) ^[1380]

6.10.1.3.4.1 Shape Attributes

Syntax:

attribute "=" *value* ";"

Example


```

shape main
{
  //Initialisation attributes - must be before drawing commands
  noshadow = "true";
  h_align = "center";

  //drawing commands
  rectangle(0, 0, 100, 100);
  println("foo bar");
}

```

Attributes:

Attribute Name	Type	Description	See also
bottomAnchorOffset	<i>(int,int)</i>	When creating a Shape Script for an embedded element (such as a Port), use this attribute to offset the shape from the bottom edge of its parent. For example: <i>bottomAnchorOffset=(0,-10)</i> ; move embedded element up 10 pixels from the bottom edge.	
dockable	<i>string</i>	Makes the shape default to dockable, so that it can be aligned with and joined to other elements (both other Shape Scripts and standard elements) on a diagram. You cannot reverse the dockable status with the Appearance menu option; to change the status, you must edit the Shape Script. Valid values: standard or off	Appearance Menu Section ^[837]
editableField	<i>string</i>	Defines a shape as an editable region of the element. This field impacts element shapes only, line glyphs are not supported. Valid Values: alias , name , note , stereotype	
endPointY, endPointX	<i>integer</i>	Only used for the reserved target and source shapes for connectors; this point determines where the main connector line connects to the end shapes. Default: 0 and 0	
fixedAspectRatio	<i>string</i>	Set to true to fix the aspect ratio. Do not use this if you do not want to fix the aspect ratio.	
h_Align	<i>string</i>	Affects horizontal placement of printed text and subshapes depending on the layoutType attribute. Valid values: left , center , or right	

Attribute Name	Type	Description	See also
layoutType	<i>string</i>	Determines how subshapes are sized and positioned. See the <i>Subshape Layout</i> topic for further details. Valid values: leftright , topdown , border	Subshape Layout <small>[1376]</small>
leftAnchorOffset	<i>(int,int)</i>	When creating a Shape Script for an embedded element (such as a Port), use this attribute to offset the shape from the left edge of its parent. For example: <i>leftAnchorOffset=(10,0)</i> ; move embedded element right 10 pixels from the left edge	
noShadow	<i>string</i>	Set to true to suppress the shapes shadow from being rendered. Valid values: true or false (default= false)	
orientation	<i>string</i>	Applies to decoration shapes only, to determine where the decoration is positioned within the containing element glyph. Valid values: NW , N , NE , E , SE , S , SW , W	
preferredHeight		Used by the border layoutType - north and south. Used in drawing the source and target shapes for connectors to determine how wide the line is.	
preferredWidth		Used by the border layoutType - east and west. Used by leftright layoutType, shapes where <i>scalable</i> is false to determine how much space they occupy for layout purposes.	
rightAnchorOffset	<i>(int,int)</i>	When creating a Shape Script for an embedded element (such as a Port), use this attribute to offset the shape from the right edge of its parent. For example: <i>rightAnchorOffset=(-10,0)</i> ; move embedded element left 10 pixels from the right edge.	
rotatable	<i>string</i>	Set to false to prevent rotation of the shape. This attribute is only applicable to the source and target shapes for lines glyphs. Valid values: true or false (default = true)	

Attribute Name	Type	Description	See also
scalable	<i>string</i>	Set to false to stop the shape from being relatively sized to the associated diagram glyph. Valid values: true or false (default= true)	
topAnchorOffset	<i>(int,int)</i>	When creating a Shape Script for an embedded element (such as a Port), use this attribute to offset the shape from the top edge of its parent. For example: <i>topAnchorOffset=(0,10)</i> ; move embedded element down 10 pixels from the top edge.	
v_Align	<i>string</i>	Affects vertical placement of printed text and subshapes depending on the layoutType attribute. Valid values: top , center , or bottom	

6.10.1.3.4.2 Drawing Methods

Syntax:

```
<MethodName> "(" <ParameterList> ")";
```

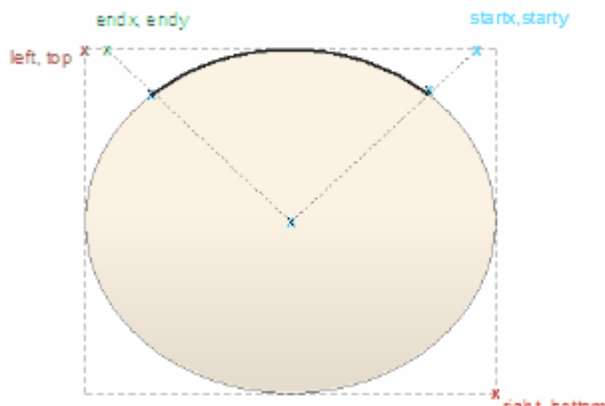
Example

```
shape main
{
  //Initialisation Attributes - these must be before drawing commands
  noshadow = "true";
  h_align = "center";

  //drawing commands (Methods)
  rectangle(0,0,100,100);
  println("foo bar");
}
```

Methods:

Method Name	Description	See also
addsubshape(string shapename(int width, int height))	Adds a sub-shape with the name shapename that must be defined within the current shape definition.	
appendcompartmenttext(string)	Appends additional strings to a compartment's text. The compartment the text is added to depends on the compartment name set using setcompartmentname (see	Add Custom Compartments to Element ^[1339]

Method Name	Description	See also
	<p>below) prior to using appendcompartmenttext.</p> <p>This method must be called to have the compartment displayed.</p>	
arc(int left, int top, int right, int bottom, int startingpointx, int startingpointy, int endingpointx, int endingpointy)	<p>Draws an elliptical anticlockwise arc with the ellipse having extents at left, top, right and bottom.</p> <p>The start point of the arc is defined by the intersection of the ellipse and the line from the center of the ellipse to the point (startingpointx, startingpointy).</p> <p>The end of the arc is similarly defined by the intersection of the ellipse and the line from the center of the ellipse to the point (endingpointx, endingpointy).</p> <p>For example:</p> <pre>Ar c (0, 0, 100, 100, 95, 0, 5, 0);</pre> 	
arco(int left, int top, int right, int bottom, int startingpointx, int startingpointy, int endingpointx, int endingpointy)	<p>As for the arc method, except that a line is drawn from the current position to the starting point of the arc, and then the current position is updated to the end point of the arc.</p>	
bezierto(int controlpoint1x, int	<p>Draws a bezier curve and updates the pen position.</p>	

Method Name	Description	See also
controlpoint1y, int controlpoint2x, int controlpoint2y, int endpointx, int endpointy)		
defSize(int width, int height)	<p>Sets the default size of the element.</p> <p>This can appear in IF and ELSE clauses with different values in each, and causes the element to be resized automatically each time the values change</p> <pre> if (HasTag(" horizontal ", " true")) { def Si ze(100, 20); r ect angl e(0, 0, 100, 100); } el se { def Si ze(20, 100); r ect angl e(0, 0, 100, 100); } </pre> <p>The above example sets the shape to the specified default size each time the Tagged Value <i>horizontal</i> is changed.</p> <p>When this is set, (Alt+Z) also resizes the shape to the defined dimensions.</p> <p>The minimum value for both int width and int height is 10.</p>	
drawnativeshape()	<p>Causes Enterprise Architect to render the shape using its usual, non-Shapescript notation; subsequent drawing commands are super-imposed over the native notation.</p> <p>This method is only enabled for element Shape Scripts; line Shape Scripts are not supported.</p>	
ellipse(int left, int top, int right, int bottom)	<p>Draws an ellipse with extents defined by left, top, right and bottom.</p>	
endpath()	<p>Ends the sequence of drawing commands that define a path.</p>	
fillandstrokepath()	<p>Fills the previously defined path with the current fill color, then draws its outline with the current pen.</p>	

Method Name	Description	See also
fillpath()	Fills the previously defined path with the current fill color.	
hidelabel(string labelname)	Hides the label specified by labelname , where labelname is one of the following values: <ul style="list-style-type: none"> • middletoplabel • middlebottomlabel • lefttoplabel • leftbottomlabel • righttoplabel • rightbottomlabel 	
image(string imageld, int left, int top, int right, int bottom)	Draws the image that has the name imageld in the Image Manager. The image must exist within the model in which the stereotype is used; if it does not already exist in the model, you must import it as reference data or select it from within a technology file. If the image is in a technology file, it should have a filename of the format <i><technology ID>::<imagename>.<extension></i> .	Reference Data ¹³⁰⁸ Adding Images in MDG Technology ¹³³³
lineto(int x, int y)	Draws a line from the current cursor position to a point specified by x and y , and then updates the pen cursor to that position.	
moveto(int x, int y)	Moves the pen cursor to the point specified by x and y .	
polygon(int centerx, int centery, int numberofsides, int radius, float rotation)	Draws a regular polygon with center at the point (centerx , centery), and numberofsides number of sides.	
print(string text)	Prints the specified text string. You cannot change the font size, type or color of this text.	
printifdefined(Prints the <i>truepart</i> if the given property exists and has a	

Method Name	Description	See also
<pre> string propertyname, string truepart (, string falsepart)) </pre>	<p>non-empty value, otherwise prints the optional <i>falsepart</i>.</p> <p>You cannot change the font size, type or color of this text.</p>	
<pre> println(string text) </pre>	<p>Appends a line of text to the shape and a line break.</p> <p>You cannot change the font size, type or color of this text.</p>	
<pre> printwrapped(string text) </pre>	<p>Prints the specified text string, wrapped over multiple lines if the text is wider than its containing shape.</p> <p>You cannot change the font size, type or color of this text.</p>	
<pre> rectangle(int left, int top, int right, int bottom) </pre>	<p>Draws a rectangle with extents at left, top, right, bottom. Values are percentages.</p>	
<pre> roundrect(int left, int top, int right, int bottom, int abs_cornerwidth, int abs_cornerheight) </pre>	<p>Draws a rectangle with rounded corners, with extents defined by left, top, right and bottom.</p> <p>The size for the corners is defined by abs_cornerwidth and abs_cornerheight; these values do not scale with the shape.</p>	
<pre> setcompartmentname(string) </pre>	<p>Sets a compartment name to the string provided.</p> <p>This method must be used <i>before</i> calling appendcompartmenttext (above); calling this <i>after</i> calling appendcompartmenttext clears any text that has already been added to the compartment.</p>	Add Custom Compartments to Element ^[1339]
<pre> setdefaultcolors() </pre>	<p>Returns the brush and pen color to the default settings, or to the user-defined colors if available.</p>	Color Queries ^[1372]
<pre> setfillcolor(int red, int green, int blue) setfillcolor(Color newColor) </pre>	<p>Sets the fill color.</p> <p>You can specify the required color by defining RGB values or using a color value returned by any of the Color Queries such as:</p> <p>Get User Fill Color ()</p>	Color Queries ^[1372]

Method Name	Description	See also
setfixedregion (int xStart, int yStart, int xEnd, int yEnd)	Fixes a region in a connector into which a subshape can be drawn, so that the subshape is not rescaled with the length or orientation of the connector line. For an example, see the end of the table in the <i>Example Scripts</i> topic.	Example Scripts <small>[1380]</small>
setlinestyle (string linestyle)	Changes the stroke pattern for commands that use the pen. string linestyle: has the following valid styles: <ul style="list-style-type: none"> • solid • dash • dot • dashdot • dashdotdot 	
setorigin (string relativeTo, int xOffset, int yOffset)	Positions floating text labels relative to the main shape. <ul style="list-style-type: none"> • relativeTo is one of N, NE, E, SE, S, SW, W, NW, CENTER • xOffset and yOffset are in pixels, not percentage values, and can be negative 	
setpen (int red, int green, int blue(, int penwidth))	Sets the pen to the defined color and optionally sets the pen width. This method is only for line-drawing commands. It does not affect any text commands.	
setpencolor (int red, int green, int blue) setpencolor (Color newColor)	Sets the pen color. You can specify the required color by defining RGB values or using a color value returned by any of the Color Queries Get User Fill Color () This method is only for line-drawing commands. It does not affect any text commands.	Color Queries <small>[1372]</small>
setpenwidth (int penwidth)	Sets the width of the pen. Pen width should be between 1 and 5 This method is only for line-drawing commands. It does not affect any text commands.	

Method Name	Description	See also
showlabel(string labelname)	Reveals the hidden label specified by labelname , where labelname is one of the following values: <ul style="list-style-type: none"> • middletoplabel • middlebottomlabel • lefttoplabel • leftbottomlabel • righttoplabel • rightbottomlabel 	
startcloudpath(puffWidth, puffHeight, noise)	Similar to StartPath , except that it draws the path with cloud-like curved segments (<i>puffs</i>). Parameters: <ul style="list-style-type: none"> • <i>float</i> puffWidth (default = 30), the horizontal distance between puffs • <i>float</i> puffHeight (default = 15), the vertical distance between puffs • <i>float</i> noise (default = 1.0), the randomization of the puffs' positions 	
startpath()	Starts the sequence of drawing commands that define a path.	
strokepath()	Draws the outline of the previously defined path with the current pen.	

6.10.1.3.4.3 Color Queries

Color queries can only be used to retrieve arguments for the *SetPenColor* and *SetFillColor* commands. These queries can be used in place of the arguments.

- **getUserFillColor()**
- **getUserBorderColor()**
- **getUserFontColor()**
- **getUserPenSize()**

```

shape main
{
    set fill color ( get user border color ( ) );
    set pen color ( get user fill color ( ) );

    rectangle ( 0, 0, 100, 100 );
}

```

6.10.1.3.4.4 Conditional Branching

Shape Scripts provide condition branching with the *if else* statement, and query methods that evaluate to either *True* or *False*.

Learn more

- [Syntax Grammar](#)^[1375] for IF statement syntax
- [Query Methods](#)^[1375] for methods that can be used as the conditional expression for IF statements
- [Example Scripts](#)^[1380] for an example

6.10.1.3.4.5 Query Methods

Two query methods are available for seeing if the associated element has certain tags or properties; these methods can be used as the conditional expression for an *if else* statement.

Method	Description	See also
boolean HasTag(string tagname, (string tagvalue))	Returns true if the associated element has a tag value with the name <i>tagname</i> . If the second parameter <i>tagvalue</i> is provided, the tag <i>tagname</i> must be present, and the value of the tag has to be equal to <i>tagvalue</i> for the method to return true .	
boolean HasProperty(string propertyname, (string propertyvalue))	Returns true if the associated element has a property with the name <i>propertyname</i> . If the second parameter <i>propertyvalue</i> is provided, the property must be present, and the value of the property has to be equal to <i>propertyvalue</i> for the method to return true .	Display Element/Connector Properties ^[1373]

6.10.1.3.4.6 Display Element/Connector Properties

The commands **print**, **println**, and **printwrapped** all take a string parameter representing the text to be printed. Element and connector properties can be added to the text using the substitution macro *#propertyname#*; for example:

```
println("name: #NAME#");
```

In addition to the properties listed below, Tagged Values can also be displayed by prefixing the Tagged Value name with *TAG*:

```
print("#TAG:condition#");
```

You can also test against and display an element's *custom* properties in the same way as you do the named properties

```
if(hasproperty("Name","Value"))
```

```
...
```

and:

```
print("#Name#");
```

There are further examples of Shape Scripts and commands in the *Example Scripts* topic.

Properties for Element Shape Scripts

- addin (value returned from an Add-In)
- alias
- author
- cardinality
- classifier
- classifier.alias
- classifier.metatype
- classifier.name
- classifier.stereotype
- classifier.type
- complexity
- concurrency
- datecreated
- datemodified
- diagram.handdrawn
- diagram.mdgtype
- diagram.name
- diagram.stereotype
- diagram.type
- haslinkeddokument
- isabstract
- isactive
- iscomposite
- isembedded
- isinparent
- isleaf
- islocked
- isroot
- isspec
- istagged
- keywords
- language
- metatype
- multiplicity
- name

- notes
- packagename
- parentedge ("right", "left", "top", "bottom")
- parent.metatype
- partition (returns "vertical" or "horizontal")
- persistence
- phase
- propertytype
- propertytype.alias
- propertytype.metatype
- propertytype.name
- propertytype.stereotype
- scope
- status
- stereotype
- subtype
- type
- version
- visibility

Properties for Connector Shape Scripts

- addin (value returned from an Add-In)
- alias
- diagram.connectornotation
- diagram.handdrawn
- diagram.mdgtype
- diagram.name
- diagram.stereotype
- diagram.type
- direction
- effect
- guard
- isroot
- isleaf
- name
- rotationdirection ("up", "down", "left", "right")
- source.aggregation

- source.alias
- source.changeable
- source.constraints
- source.element.name
- source.element.stereotype
- source.metatype
- source.multiplicity
- source.multiplicityisordered
- source.name
- source.qualifiers
- source.stereotype
- source.targetscope
- stereotype
- target.aggregation
- target.alias
- target.changeable
- target.constraints
- target.element.name
- target.element.stereotype
- target.metatype
- target.multiplicity
- target.multiplicityisordered
- target.name
- target.qualifiers
- target.stereotype
- target.targetscope
- type
- weight

6.10.1.3.4.7 Sub-Shapes

Shapes can contain - and be composed of - other shapes.

Subshape Layout:

To set the layout type, the *layoutType* attribute must be set in the *initialization attributes* section of the script; in other words, before any of the methods are called. Valid values for this attribute are:

- **LeftRight** - Shapes with leftright layout position subshapes side by side, with the first added on the left, and subsequent subshapes to the right
- **TopDown** - TopDown places subshapes in a vertical arrangement, with the first shape added to the top and subsequent shapes added below

- **Border** - Border layout requires an additional argument to the *addsubshape* method to specify which region of the containing shape the subshape is to occupy: N, E, S, W or CENTER; each region can only be occupied by one subshape

A subshape that is assigned to the E or W region must have its *preferredwidth* attribute specified in its declaration and, similarly, subshapes added to N or S must have their *preferredheight* attribute set; in this case, the values for these attributes are treated as static lengths and do not scale glyphs

For example:

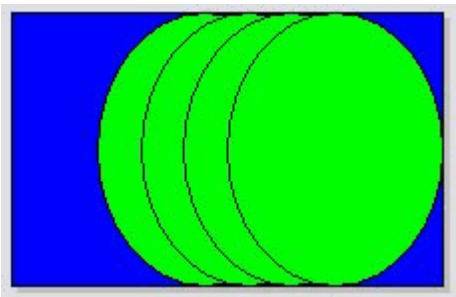
```

shape main
{
  layout type="topdown";
  setfillcolor(0, 0, 255);
  rectangle(0, 0, 100, 100);
  addsubshape("sub", 50, 100, 20, 0);
  addsubshape("sub", 50, 100, 30, -100);
  addsubshape("sub", 50, 100, 40, -200);
  addsubshape("sub", 50, 100, 50, -300);

  shape sub
  {
    setfillcolor(0, 255, 0);
    ellipse(0, 0, 100, 100);
  }
}

```

The above script provides the following shape:



6.10.1.3.4.8 Reserved Names

Elements:

Elements (such as Class, State or Event) have the following reserved names for parts of the shape.

Name	Description
shape main	The main shape is the whole shape.
shape label	The shape label gives the shape a detached label.
decoration <identifier>	Decoration gives the shape a decoration as defined by the

Name	Description
	name in <identifier>.

Connectors:

Connectors (such as Association, Dependency or Generalization) have the following reserved names for parts of the shape.

Name	Description
shape main	The main shape is the whole shape.
shape source	The source shape is an extra shape at the source end of the connector.
shape target	The target shape is an extra shape at the target end of the connector.
shape <labelID>	The <labelID> gives the connector a detached label, where <labelID> is one of the following: <ul style="list-style-type: none"> • LeftTopLabel • MiddleTopLabel • RightTopLabel • LeftBottomLabel • MiddleBottomLabel • RightBottomLabel

6.10.1.3.4.9 Miscellaneous**Return Command:**

Execution of the script can be terminated by using the return command. See [Example Scripts](#)^[1380] for an example.

Looping:

The Shape Script feature does not support looping constructs.

Comments:

C-style comments are supported.

```
// C Style Single Line comment
/* Multi Line
```

comment supported */

String Manipulation:

Not supported.

Arithmetical Operations:

Not supported.

Variables Declaration:

Not supported.

Change ShapeScript Fonts:

Not possible.

Can I apply a Shapescrpt without using Stereotypes?

No.

6.10.1.3.4.10 Syntax Grammar

Grammar symbols:

- * = zero or more
- + = one or more
- | = or
- ; = terminator

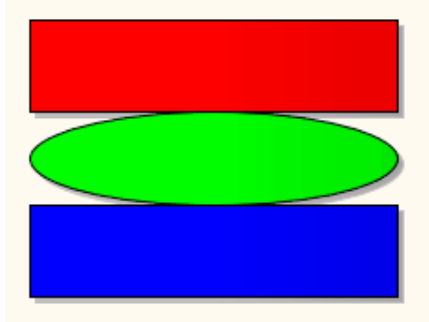
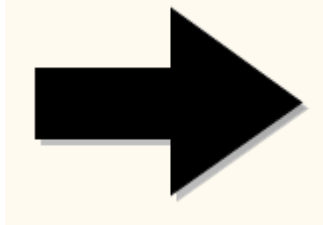
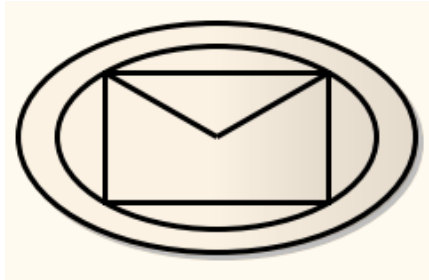
ShapeScript	::=	<Shape>*;
Shape	::=	<ShapeDeclaration> <ShapeBody>;
ShapeDeclaration	::=	<ShapeType> <ShapeName>;
ShapeType	::=	"shape" "decoration";
ShapeName	::=	<ReservedShapeName> <stringliteral>;
ReservedShapeName	::=	See Reserved Names ¹³⁷⁷ for full reserved shape listing.


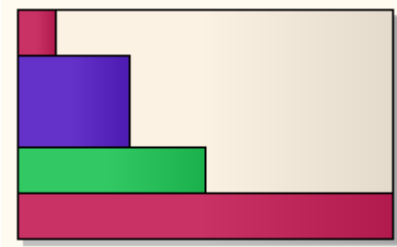
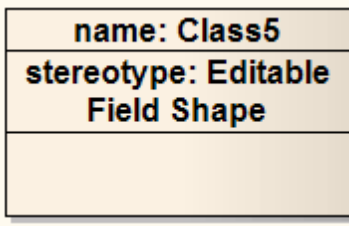
ShapeBody	::=	"{" <InitialisationAttributeAssignment>* <DrawingStatement>* <SubShape>* "}";
InitialisationAttributeAssignment	::=	<Attribute> "=" <Value> ";";
Attribute	::=	See Shape Attributes ^[1363] for full listing of attribute names.
DrawingStatement	::=	<IfElseSection> <Method>;
IfElseSection	::=	"if" "(" <QueryExpression> ")" <TrueSection> (<ElseSection>) ;
QueryExpression	::=	<QueryName> "(" <ParameterList> ")";
QueryName	::=	See Query Methods ^[1373] for a full listing of Query names.
TrueSection	::=	"{" <DrawingStatement>* "}"
ElseSection	::=	"else" "{" <DrawingStatement>* "}"
Method	::=	<MethodName> "(" <ParameterList> ")" ";";
MethodName	::=	See Drawing Methods ^[1366] for a full listing of method names.

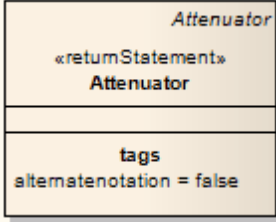
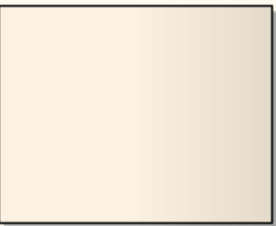
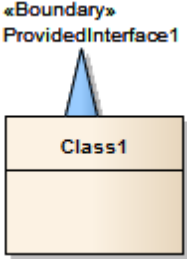
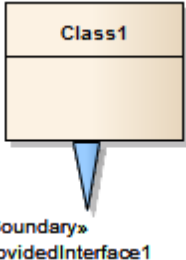
6.10.1.3.5 Example Scripts

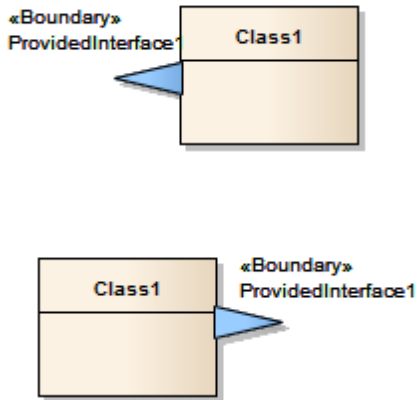
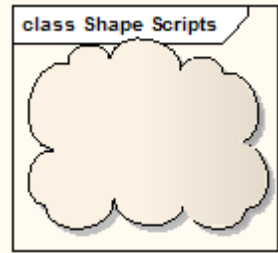
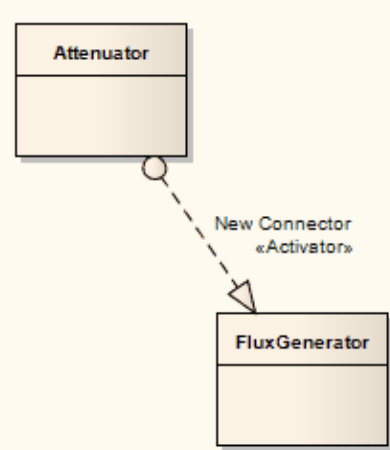
Below is a selection of example Shape Scripts.

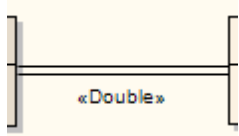
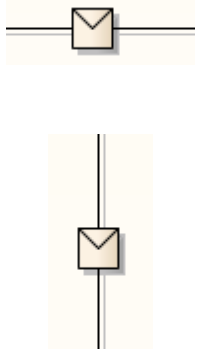
Code	Result
Code to add compartments for child elements, with and without editing the parent element.	See Add Custom Compartments to Element ^[1339]
<pre>// BASIC SHAPES shape main { setfillcolor(255,0,0); // (R,G,B) rectangle(0,0,90,30); // (x1,y1,x2,y2) setfillcolor(0,255,0); // (R,G,B) ellipse(0,30,90,60); // (x1,y1,x2,y2)</pre>	

Code	Result
<pre> setfillcolor(0,0,255); // (R,G,B) rectangle(0,60,90,90); // (x1,y1,x2,y2) } </pre>	
<pre> // SINGLE CONDITIONAL SHAPE shape main { if (HasTag("Trigger","Link")) { // Only draw if the object has a Tagged Value // Trigger=Link // Set the fill color for the path setfillcolor(0,0,0); startpath(); // Start to trace out a path moveto(23,40); lineto(23,60); lineto(50,60); lineto(50,76); lineto(76,50); lineto(50,23); lineto(50,40); endpath(); // End tracing out a path // Fill the traced path with the fill color fillandstrokepath(); return; } } </pre>	
<pre> // MULTI CONDITIONAL SHAPE shape main { startpath(); ellipse(0,0,100,100); endpath(); fillandstrokepath(); ellipse(3,3,27,27); if (HasTag("Trigger","None")) { return; } if (HasTag("Trigger","Error")) { setfillcolor(0,0,0); startpath(); moveto(23,77); lineto(37,40); lineto(60,47); lineto(77,23); lineto(63,60); } } </pre>	

Code	Result
<pre> lineto(40,53); lineto(23,77); endpath(); fillandstrokepath(); return; } if (HasTag("Trigger","Message")) { rectangle(22,22,78,78); moveto(22,22); lineto(50,50); lineto(78,22); return; } } </pre>	
<pre> // SUB SHAPES shape main { rectangle(0,0,100,100); addsubshape("red", 10, 20); addsubshape("blue", 30, 40); addsubshape("green", 50, 20); addsubshape("red", 100, 20); shape red { setfillcolor(200, 50, 100); rectangle(0,0,100,100); } shape blue { setfillcolor(100, 50, 200); rectangle(0,0,100,100); } shape green { setfillcolor(50, 200, 100); rectangle(0,0,100,100); } } </pre>	
<pre> // EDITABLE FIELD SHAPE shape main { rectangle(0,0,100,100); addsubshape("namecompartment", 100, 20); addsubshape("stereotypercompartment", 100, 40); shape namecompartment { h_align = "center"; editablefield = "name"; rectangle(0,0,100,100); println("name: #name#"); } } </pre>	

Code	Result
<pre> } shape stereotypecompartment { h_align = "center"; editablefield = "stereotype"; rectangle(0,0,100,100); println("stereotype: #stereotype#"); } } </pre>	
<pre> // RETURN STATEMENT SHAPE shape main { if(hasTag("alternatenotation", "false")) { //draw ea's inbuild glyph drawnativeshape(); //exit script with the return statement return; } //alternate notation commands //... rectangle(0,0,100,100); } </pre>	 <p>The diagram shows a class box for 'Attenuator'. The top compartment contains the text '«returnStatement» Attenuator'. The bottom compartment contains a 'tags' section with the value 'alternatenotation = false'.</p> 
<pre> //EMBEDDED ELEMENT SHAPE POSITION ON PARENT EDGE shape main { defsize(60,60); startpath(); if(hasproperty("parentedge", "top")) { moveto(0,100); lineto(50,0); lineto(100,100); } if(hasproperty("parentedge", "bottom")) { moveto(0,0); lineto(50,100); lineto(100,0); } if(hasproperty("parentedge", "left")) { moveto(100,0); lineto(0,50); lineto(100,100); } if(hasproperty("parentedge", "right")) </pre>	 <p>The diagram shows a class box for 'Class1'. A blue triangle points upwards from the top edge of the box. Above the box, the text '«Boundary» ProvidedInterface1' is displayed.</p>  <p>The diagram shows a class box for 'Class1'. A blue triangle points downwards from the bottom edge of the box. Below the box, the text '«Boundary» ProvidedInterface1' is displayed.</p>

Code	Result
<pre> { moveto(0,0); lineto(100,50); lineto(0,100); } endpath(); setfillcolor(153,204,255); fillandstrokepath(); } </pre>	
<pre> // CLOUD PATH EXAMPLE SHAPE shape main { StartCloudPath(); Rectangle(0,0,100,100); EndPath(); FillAndStrokePath(); } </pre>	
<pre> // CONNECTOR SHAPE shape main { // draw a dashed line noshadow=true; setlinestyle("DASH"); moveto(0,0); lineto(100,0); } shape source { // draw a circle at the source end rotatable = true; startpath(); ellipse(0,6,12,-6); endpath(); fillandstrokepath(); } shape target { // draw an arrowhead at the target end rotatable = true; startpath(); moveto(0,0); lineto(16,6); lineto(16,-6); endpath(); } </pre>	

Code	Result
<pre>fillandstrokepath(); }</pre>	
<pre>// DOUBLE LINE shape main { setlinestyle("DOUBLE"); moveto(0,0); lineto(100,0); }</pre>	
<pre>// ROTATION DIRECTION shape main { moveto(0,0); lineto(100,0); setfixedregion(40,-10,60,10); rectangle(40,-10,60,10); if(hasproperty("rotationdirection","up")) { moveto(60,-10); lineto(50,0); lineto(60,10); } if(hasproperty("rotationdirection","down")) { moveto(40,-10); lineto(50,0); lineto(40,10); } if(hasproperty("rotationdirection","left")) { moveto(40,-10); lineto(50,0); lineto(60,-10); } if(hasproperty("rotationdirection","right")) { moveto(40,10); lineto(50,0); lineto(60,10); } }</pre>	

6.10.1.4 Tagged Value Types

Enterprise Architect provides a number of predefined Tagged Value Types that enable you to create your own:

- Tagged Values that are structured with a specific format, with or without tag filters, or
- Tagged Values that return values from the various reference data tables

You can also use a masking parameter to create your own customized masked Tagged Value Type.

Notes

- You can transport Tagged Value Type definitions between models, using the Export Reference Data and Import Reference Data options; Tagged Value Types are exported as *Property Types*

Learn more

- [Predefined Tagged Value Types](#)^[1386]
- [Predefined Tagged Value Type Filters](#)^[1389]
- [Custom Tagged Value Type](#)^[1393]
- [Reference Data](#)^[1390]
- [Export Reference Data](#)^[306]
- [Import Reference Data](#)^[308]

6.10.1.4.1 Predefined Structured Types

This table details the predefined structured Tagged Value types, along with the syntax used to create the initial values for their use. You use these to create your own structured Tagged Values.

Tagged Value Type and **Format** entries are case-sensitive.

Tagged Value Type	Format	Description	See also
AddinBroadcast	Type=AddinBroadcast ; ; Values=YourAddinName ;	Allows an AddIn to respond to an attempt to edit this Tagged Value by showing a dialog in which the value and notes can be edited.	
Boolean	Type=Boolean ; Default=Val ;	Enables input of True or False , either of which can be the default value.	
CheckList	Type=CheckList ; Values=Val1,Val2,Val3 ;	Enables you to add a checklist of things to be completed or satisfied before an action is approved or performed. Val1, Val2, Val3 and so on specify the checklist items, each of which is rendered in the Tagged Values window with a checkbox; the tag has the value Incomplete until each checkbox is selected. For example: Type=CheckList ; Values=Does the change solve the task\issue given,Does the code have sufficient error handling,Does the code make sense,Does the code comply with the coding conventions ;	
Classifier	Type=Classifier ; Values=Type1,Type2 ; Stereotypes=Stereotype1 ;	Deprecated - use RefGUID and RefGUIDList Returns the <i>name</i> of a user-selected element from the model, where Type1 and Type2 specify one or more allowed element types and Stereotype1 represents an	

Tagged Value Type	Format	Description	See also
		allowed stereotype.	
Color	Type=Color; Default=Val;	Enables input of a color value from a color chooser menu, where the value is the decimal integer translation of the color's Hex RGB value. For example, the RGB for Red is FF , and the decimal value is 255 .	
Const	Type=Const; Default=Val;	Enables creation of a read-only constant value.	
Custom	Type= Custom;	Enables you to create your own template for predefined types.	Create Custom Tagged Value Type <small>[1393]</small>
DateTime	Type=DateTime;	Enables input of the date and time for the Tagged Value from a calendar menu.	
Directory	Type=Directory; Default=Val;	Enables entry of a directory path from a browser. You can set a default directory path as a string value.	
Enum	Type=Enum; Values=Val1,Val2,Val3; Default=Val2;	Enables definition of a comma-separated list, where Val1 , Val2 and Val3 represent values in the list and Default represents the default value of the list.	
File	Type=File; Default=Val;	Enables input of a filename from a file browser dialog. The named file can be launched in its default application. You can set a default file as a string containing the file path and file name.	
Float, Decimal, Double	Type=Float; Type=Decimal; Type=Double; Default=Val;	Enables entry of a Float, Decimal or Double value. These types all map to the same type of data. You can set a default for any or all of these.	
Integer	Type=Integer; Default=Val;	Enables entry of an Integer value, and a default.	

Tagged Value Type	Format	Description	See also
Memo	Type=Memo;	Enables input of large and complex Tagged Values.	
RefGUID	Type=RefGUID; Values=Type1,Type2; Stereotypes=Stereotype1;	<p>Enables the Tagged Value to reference an element from the model by specifying the element's <i>GUID</i>, where:</p> <ul style="list-style-type: none"> • Type1 and Type2 specify one or more allowed diagram objects (such as Class, Component, Attribute or Operation) • Stereotype1 represents an allowed stereotype <p>You set the classifier, attribute or operation for a Tagged Value of this type by clicking on the (...) button against the Tagged Value in the Tagged Value window.</p>	Classifier ^[880] , Attribute or Operation ^[884]
RefGUIDList	Type=RefGUIDList; Values=Type1,Type2; Stereotypes=Stereotype1;	<p>Enables the Tagged Value to reference a list of elements from the model by specifying each element's <i>GUID</i>, where:</p> <ul style="list-style-type: none"> • Type1 and Type2 specify one or more allowed diagram objects (such as Class or Component) • Stereotype1 represents an allowed stereotype <p>You set the classifier, attribute or operation for a Tagged Value of this type by clicking on the (...) button against the Tagged Value in the Tagged Value window.</p>	Classifier ^[880] , Attribute or Operation ^[884]
Spin	Type=Spin; LowerBound=x; UpperBound=x; Default=Val;	<p>Enables creation of a spin control with the value of LowerBound being the lowest value and UpperBound being the highest value.</p> <p>You can also set a default within that range.</p>	
String	Type=String; Default=Val;	<p>Enables entry of a string value, up to 255 characters in length, and a default text string.</p> <p>For longer texts, use Type=Memo.</p>	Type= Memo ^[1388]

Tagged Value Type	Format	Description	See also
URL	Type=URL; Default=Val;	Enables entry of a web URL. The URL should start with: <ul style="list-style-type: none"> • 'http://' • 'https://' • 'www.' <p>You can set a default URL as a string value.</p>	

Tag Filters:

The following table details filters that can be used to restrict where a Tagged Value can be applied.

Filter	Format	Description	See also
AppliesTo	AppliesTo=Type1,Type2;	Restricts the element types this filter can be applied to, where Type1 and Type2 are the valid types. Possible values are: <ul style="list-style-type: none"> • All element types • All connector types • Attribute • Operation, and • OperationParameter 	
BaseStereotype	BaseStereotype=S1,S2;	Restricts the stereotypes that this tag belongs to, where S1 and S2 are the allowed stereotypes.	

Learn more

- [Create Your Own Structured Tagged Values](#)¹³⁸⁹

6.10.1.4.2 Create Structured Tagged Values

You can create your own Tagged Value based on a predefined structured Tagged Value Type.

Access **Settings | UML Types > Tagged Value Types**

Reference

Complete the fields on the Tagged Value Types tab, as indicated below:

Field/Button	Description	See also
New	Click to enable the fields to enter information for a new Tagged Value.	
Tag Name	Type the appropriate Tagged Value name.	
Description	If required, type the purpose of the Tagged Value.	
Detail	Either copy-and-paste or type the syntax of the predefined structured Tagged Value Type.	
Save	Click to save the new structured Tagged Value. The Tagged Value type displays in the Defined Tag Types list.	

Learn more

- [Predefined Structured Types](#)^[1386]

6.10.1.4.3 Predefined Reference Data Types

This table details the predefined Reference Data Tagged Value types that are used to return the values held in a relevant table in Enterprise Architect, along with the syntax required for their use.

You use these to create your own Reference Data Tagged Values.

Tagged Value Type	Format	Drop-Down List Returned	See also
Authors	Type=Enum; List=Authors;	Authors that have been defined for the Enterprise Architect model.	
Cardinality	Type=Enum; List=Cardinality;	Cardinality types that have been defined for the Enterprise Architect model.	
Clients	Type=Enum; List=Clients;	Clients that have been defined for the Enterprise Architect model.	
ComplexityTypes	Type=Enum; List=ComplexityTypes;	Complexity types that have been defined for the Enterprise Architect model. Whilst complexity types can be exported and imported as project reference data,	Reference Data ^[987]

Tagged Value Type	Format	Drop-Down List Returned	See also
		they cannot be updated and so are effectively standard across all projects.	
ConstraintTypes	Type=Enum; List=ConstraintTypes;	Constraint types that have been defined for the Enterprise Architect model.	
EffortTypes	Type=Enum; List=EffortTypes;	Effort types that have been defined for the Enterprise Architect model.	
MaintenanceTypes	Type=Enum; List=MaintenanceTypes;	Maintenance types that have been defined for the Enterprise Architect model.	
ObjectTypes	Type=Enum; List=ObjectTypes;	Object types that have been defined for the Enterprise Architect model.	
Phases	Type=Enum; List=Phases;	Phases that have been defined for the Enterprise Architect model.	
ProblemTypes	Type=Enum; List=ProblemTypes;	Problem types that have been defined for the Enterprise Architect model.	
RoleTypes	Type=Enum; List=RoleTypes;	Role types that have been defined for the Enterprise Architect model.	
RequirementTypes	Type=Enum; List=RequirementTypes;	Requirement types that have been defined for the Enterprise Architect model.	
Resources	Type=Enum; List=Resources;	Resources that have been defined for the Enterprise Architect model.	
RiskTypes	Type=Enum; List=RiskTypes;	Risk types that have been defined for the Enterprise Architect model.	
RTFTemplates	Type=Enum; List=RTFTemplates;	RTF Templates that have been defined for the Enterprise Architect model.	
ScenarioTypes	Type=Enum; List=ScenarioTypes;	Scenario types that have been defined for the Enterprise Architect model.	
TestTypes	Type=Enum;	Test types that have been defined for the	

Tagged Value Type	Format	Drop-Down List Returned	See also
	List=TestTypes;	Enterprise Architect model.	

Learn more

- [Create Reference Data Tagged Values](#) ^[1392]

6.10.1.4.4 Create Reference Data Tagged Values

You can create your own Tagged Value based on a predefined Reference Data Tagged Value Type.

Access [Settings | UML Types > Tagged Value Types](#)

Reference

Complete the fields on the Tagged Value Types tab, as indicated below:

Field/Button	Description	See also
New	Click to enable the fields to enter information for a new Tagged Value.	
Tag Name	Type the appropriate Tagged Value name.	
Description	If required, type the purpose of the Tagged Value.	
Detail	Either copy-and-paste or type the syntax of the predefined Reference Data Tagged Value Type.	Model Elements and Features with Tagged Values ^[970]
Save	Click to save the new Reference Data Tagged Value. The Tagged Value type displays in the Defined Tag Types list.	

Notes

- If the values in the reference data are changed after the Tagged Value Type is created, Enterprise Architect must be reloaded in order to reflect the changes in the Tagged Value Type

Learn more

- [Predefined Reference Data Types](#) ^[1390]

6.10.1.4.5 Create Custom Tagged Value Type

Custom masked Tagged Values give you great flexibility in designing model components that accept data entries.

Access [Settings | UML Types > Tagged Value Types](#)

Reference

On the Tagged Value Types tab:

Field	Action	See also
Tag	Type an appropriate name for the Tagged Value.	
Description	Type the purpose of the Tagged Value, if required.	
Detail	Type Type=Custom;	

The type **Custom** enables you to set up the appropriate mask, using the following characters to define the format of the mask:

Mask	Description
D	Enables the Tagged Value to display digits only.
d	Enables the Tagged Value to display digits or spaces.
+	Enables the use of + , - or <i>spaces</i> .
C	Enables the use of alpha characters only.
c	Enables the Tagged Value to be an alpha character or a space.
A	Enables the use of alphanumeric characters.
a	Enables the Tagged Value to use alphanumeric values or a space.

In the diagram below, the **Mask** configuration option shows syntax that first defines seven blank spaces, which are occupied by characters determined by the template option.

The first two visible characters in the **Mask** option are represented by a lower case **c** indicating that the enableable information can be entered as either an alpha character or as a space.

The following blank spaces again indicate space defined by the template option, and the remaining characters are defined by the **d** character, which represents the enableable characters as digits or spaces. The hyphen is present in the final output, splitting up the digits.

With the **Template** configuration option, the syntax defines the template of the masked option by occupying the blank spaces that are present in the **Mask** option. The template is used to ensure that this information is present with every use of this custom Tagged Value. The underscored values indicate the area that is to be occupied by data input by the user as defined in the **Mask** option.

Type	Description
IORules	IORules
MemberZip	Zip Code
Message	
OutMessage	

6.10.1.5 Code Template Framework

The Code Template Framework (CTF) is used for:

- Forward engineering of UML models
- Behavioral Code generation
- Model Driven Architecture (MDA) Transformations

This section discusses how you can customize the way in which Enterprise Architect:

- Generates source code, using the Code Template Editor, and
- Transforms models, using the Transformation Template Editor

Topics

Topic	Detail	See also
Code Templates	<p>When forward engineering a Class model, the code templates define how the skeletal code is to be generated for a given programming language.</p> <p>The templates are written as plain text with a syntax that shares some aspects of both mark-up languages and scripting languages.</p> <p>Enterprise Architect provides a set of Base Templates that you can add to or adapt if required.</p>	<p>Code Template Framework^[1924]</p> <p>Code Template Syntax^[1406]</p> <p>Base Templates^[1397]</p> <p>The Code Template Editor in MDG Development^[1462]</p>
Model Transformation Templates	<p>Model Transformation Templates provide a fully configurable method of defining how Model Driven Architecture (MDA) Transformations convert model elements and model fragments from one domain to another.</p> <p>This process is two-tiered. It creates an intermediary language (which can be viewed for debugging) which is then processed to create the objects.</p>	<p>Transformation Templates^[1710]</p> <p>Write Transformations^[1712]</p> <p>Intermediary Language^[1714]</p>
Behavioral Code Generation Templates	<p>Enterprise Architect supports user-definable code generation of the UML behavioral models.</p> <p>This uses the standard Code Template Framework but includes specific Enterprise Architect Simulation Library (EASL) code generation macros.</p>	<p>The Code Template Editor in MDG Development^[1462]</p> <p>EASL Code Generation Macros^[1449]</p>

Learn more

- [Code Template Editor](#)^[1462]
- [Macros](#)^[1409]

6.10.1.5.1 Code and Transform Templates

Topics

Topic	Detail	See also
Usage	Code and transform templates enable you to customize code generation of existing languages or customize an MDA Transformation.	<p>Generate Source Code^[1768]</p> <p>Model Transformation^[1679]</p>
Start Point	By default the <i>File</i> template is the start point of a process though the templates.	<p>Base Templates^[1397]</p>

	<p>Calls can be made to the predefined templates as well as to user defined templates.</p> <p>The order of the predefined templates in the Code Template Editor and Transformation Template Editor relates to the hierarchical order of the objects and their parts that are to be processed</p> <p><i>File</i> consists of <i>Classes</i> which can contain <i>Attributes</i> and <i>Operations</i>.</p>	Built-in Transformations ^[1686]
Calling Templates	<p>Within any template, other templates can be called using %TemplateName%. The enclosing percent (%) signs indicate a macro.</p> <p>You would use this for a single call to the ClassBody template, %ClassBody%, as below:</p> <pre>%list="TemplateName" @separator="\n" @indent=" "%</pre> <p>The %list macro performs an iterative pass on all the objects in the scope of the current template and calls the <i>TemplateName</i> for each of these:</p> <pre>%list="ClassBody" @separator="\n" @indent=" "%</pre> <p>After generation or transformation, each macro is substituted to produce the generated output; for a language such as C++, the result of processing the above template might be:</p> <pre>/** * This is an example class note generated using * code templates * @author Sparx Systems */ class ClassA: public ClassB { ... }</pre>	List Macro ^[1444]
Execution of Code Templates	<p>Each template is designed for use with a particular element; for example, the <i>ClassNotes</i> template is to be used with UML Class elements.</p> <p>The element that is currently being generated is said to be <i>in scope</i>; if the element in scope is stereotyped, Enterprise Architect looks for a template that has been defined for that stereotype.</p> <p>If a match is found, the specialized template is executed; otherwise the default implementation of the base template is used.</p> <p>Templates are processed sequentially, line by line, replacing each macro with its underlying text value from the model.</p>	
Transfer Templates Between Projects	<p>If you edit a base Code Generation or Transformation template, or create a customized template, you can copy them from one project to another as Reference Data.</p>	Export Code Generation and Transformation Templates ^[1400] Import Code Generation

		and Transformation Templates <small>[1401]</small>
--	--	---

6.10.1.5.1.1 Base Templates

The Code Template Framework consists of a number of base templates. Each base template transforms particular aspects of the UML to corresponding parts of object-oriented languages.

Topics

Topic	Detail	See also
Further Information	<p>The base templates form a hierarchy, which varies slightly across different programming languages.</p> <p>In a typical template hierarchy relevant to a language like C# or Java (which do not have header files) the templates can be modeled as Classes, but usually are just plain text.</p> <p>This hierarchy would be slightly more complicated for languages like C++ and Delphi, which have separate implementation templates.</p> <p>Each of the base templates must be specialized to be of use in code engineering; in particular, each template is specialized for the supported languages (or 'products').</p> <p>For example, there is a <i>ClassBody</i> template defined for C++, another for C#, another for Java, and so on; by specializing the templates, you can tailor the code generated for the corresponding UML entity.</p> <p>Once the base templates are specialized for a given language, they can be further specialized based on:</p> <ul style="list-style-type: none"> • A Class's stereotype • A feature's stereotype (where the feature can be an operation or attribute) <p>This type of specialization enables, for example, a C# operation that is stereotyped as «property» to have a different <i>Operation Body</i> template from an ordinary operation; the <i>Operation Body</i> template can then be specialized further, based on the Class stereotype.</p>	Code Template Framework <small>[1394]</small> Code and Transformation Templates <small>[1395]</small>

Reference

The following table lists and briefly describes the base templates used in the CTF.

Template	Description
Attribute	A top-level template to generate member variables from UML attributes.

Template	Description
Attribute Declaration	Used by the <i>Attribute</i> template to generate a member variable declaration.
Attribute Notes	Used by the <i>Attribute</i> template to generate member variable notes.
Class	A top-level template for generating Classes from UML Classes.
Class Base	Used by the <i>Class</i> template to generate a base Class name in the inheritance list of a derived Class, where the base Class doesn't exist in the model.
Class Body	Used by the <i>Class</i> template to generate the body of a Class.
Class Declaration	Used by the <i>Class</i> template to generate the declaration of a Class.
Class Interface	Used by the <i>Class</i> template to generate an interface name in the inheritance list of a derived Class, where the interface doesn't exist in the model.
Class Notes	Used by the <i>Class</i> template to generate the Class notes.
File	A top-level template for generating the source file. For languages such as C++, this corresponds to the header file.
Import Section	Used in the <i>File</i> template to generate external dependencies.
Linked Attribute	A top-level template for generating attributes derived from UML Associations.
Linked Attribute Notes	Used by the <i>Linked Attribute</i> template to generate the attribute notes.
Linked Attribute Declaration	Used by the <i>Linked Attribute</i> template to generate the attribute declaration.
Linked Class Base	Used by the <i>Class</i> template to generate a base Class name in the inheritance list of a derived Class, for a Class element in the model that is a parent of the current Class.
Linked Class Interface	Used by the <i>Class</i> template to generate an Interface name in the inheritance list of a derived Class, for an Interface element in the model that is a parent of the current Class.

Template	Description
Namespace	A top-level template for generating namespaces from UML packages (although not all languages have namespaces, this template can be used to generate an equivalent construct, such as packages in Java).
Namespace Body	Used by the <i>Namespace</i> template to generate the body of a namespace.
Namespace Declaration	Used by the <i>Namespace</i> template to generate the namespace declaration.
Operation	A top-level template for generating operations from a UML Class's operations.
Operation Body	Used by the <i>Operation</i> template to generate the body of a UML operation.
Operation Declaration	Used by the <i>Operation</i> template to generate the operation declaration.
Operation Notes	Used by the <i>Operation</i> template to generate documentation for an operation.
Parameter	Used by the <i>Operation Declaration</i> template to generate parameters.

The second table lists templates used for generating code for languages that have separate interface and implementation sections.

Template	Description
Class Impl	A top-level template for generating the implementation of a Class.
Class Body Impl	Used by the <i>Class Impl</i> template to generate the implementation of Class members.
File Impl	A top-level template for generating the implementation file.
File Notes Impl	Used by the <i>File Impl</i> template to generate notes in the source file.
Import Section Impl	Used by the <i>File Impl</i> template to generate external dependencies.
Operation Impl	A top-level template for generating operations from a UML Class's operations.

Template	Description
Operation Body Impl	Used by the <i>Operation</i> template to generate the body of a UML operation.
Operation Declaration Impl	Used by the <i>Operation</i> template to generate the operation declaration.
Operation Notes Impl	Used by the <i>Operation</i> template to generate documentation for an operation.

6.10.1.5.1.2 Export Code Generation and Transformation Templates

Access [Project | Model Import/Export | Export Reference Data](#)

Use to

- Transfer Code Generation and Transformation templates between models

How to

To export a Code Generation template or a Transformation template

Step	Action	See also
1	On the Export Reference Data dialog, in the Name list, select the templates to export. The list includes any standard Code Generation or Transformation templates that have been <i>changed</i> , and any customized templates that you have created or changed. You can select one or more templates to be exported to a single XML file, by pressing (Ctrl) or (Shift) as you click on the template names.	
2	Click on the Export button.	
3	When prompted to do so, enter a valid file name with a .XML extension.	
4	Click on the Save and OK buttons. This exports the template(s) to the file; you can use any text or XML viewer to examine the file.	

[Learn more](#)

- [Write Transformations](#)^[1712]
- [Import Code Generation and Transformation Templates](#)^[1401]
- [Export Reference Data](#)^[306]

6.10.1.5.1.3 Import Code Generation and Transformation Templates

Access Project | Model Import/Export | Import Reference Data

Use to

- Transfer Code Generation and Transformation templates between models

How to

To import a Code Generation template or a Transformation template

Step	Action	See also
1	On the Import Reference Data dialog, click on the Select File button and browse to the .XML file containing the required Code Generation or Transformation template.	
2	Select the name of one or more template datasets and click on the Import button.	

Learn more

- [Write Transformations](#)^[1712]
- [Import Reference Data](#)^[308]
- [Export Code Generation and Transformation Templates](#)^[1400]

6.10.1.5.1.4 Synchronize Code

Enterprise Architect uses code templates during the forward synchronization of the following programming languages:

- ActionScript
- C
- C++
- C#
- Delphi
- Java
- PHP
- Python
- VB
- VB.Net

Topics

Topic	Detail	See also
Change Types	<p>Three types of change can occur in the source when it is synchronized with the UML model:</p> <ul style="list-style-type: none"> • Synchronize Existing Sections: for example, changing the return type in an operation declaration • Add New Sections to Existing Features: for example, adding notes to a Class declaration, where there were previously none • Add New Features and Elements: for example, adding a new operation to a Class <p>Each of these changes must be handled differently by Enterprise Architect; their effect on the CTF is described in the topics listed on the right.</p>	<p>Synchronize Existing Sections ^[1403]</p> <p>Add New Sections to Existing Features ^[1403]</p> <p>Add New Features and Elements ^[1404]</p>

Only a *subset* of the CTF base templates is used during synchronization. This subset corresponds to the distinct sections that Enterprise Architect recognizes in the source code. The following table lists the code templates and the corresponding code sections that can be synchronized.

Code Template	Code Section
Class Notes	Comments preceding Class declaration.
Class Declaration	Up to and including Class parents.
Attribute Notes	Comments preceding Attribute declaration.
Attribute Declaration	Up to and including terminating character.
Operation Notes	Comments preceding operation declaration.
Operation Notes Impl	As for <i>Operation Notes</i> .
Operation Declaration	Up to and including terminating character.
Operation Declaration	Up to and including terminating character.

Code Template	Code Section
Impl	
Operation Body	Everything between and including the braces.
Operation Body Impl	As for <i>Operation Body</i> .

Learn more

- [Base Templates](#) f1397

Topics

Topic	Detail	See also
Usage	<p>When an existing section in the source code differs from the result generated by the corresponding template, that section is replaced.</p> <p>Consider for example, the following C++ Class declaration:</p> <pre>(asm) class A: public B</pre> <p>Now assume you add an inheritance relationship from Class A to Class C; the entire Class declaration would be replaced with something like:</p> <pre>(asm) class A: public B, public C</pre>	

Topics

Topic	Detail	See also
Usage	<p>The following can be added as new sections, to existing features in the source code:</p> <ul style="list-style-type: none"> • Class Notes • Attribute Notes • Operation Notes • Operation Notes Impl • Operation Body • Operation Body Impl <p>Assume Class A from the previous example had no note when</p>	<p>Synchronize Existing Sections <small>f1403</small></p>

	<p>you originally generated the code, and that you now specify a note in the model for Class A. Enterprise Architect attempts to add the new note from the model during synchronization, by executing the <i>Class Notes</i> template.</p> <p>To make room for the new section to be inserted, you can specify how much white space to append to the section via synchronization macros.</p>	Synchronization Macros ^[1447]
--	--	--

Topics

Topic	Detail	See also
Usage	<p>The following features and elements can be added to the source code during synchronization:</p> <ul style="list-style-type: none"> • Attributes • Inner Classes • Operations <p>These are added by executing the relevant templates for each new element or feature in the model.</p> <p>Enterprise Architect attempts to preserve the appropriate indenting of new features in the code, by finding the indents specified in list macros of the Class; for languages that make use of namespaces, the <i>synchNamespaceBodyIndent</i> macro is available.</p> <p>Classes defined within a (non-global) namespace are indented according to the value set for this macro, during synchronization.</p> <p>The value is ignored:</p> <ul style="list-style-type: none"> • For Classes defined within a package setup as a root namespace, or • If the Generate Namespaces option is set to False in the appropriate language page (C#, C++ or VB.Net) on the Options dialog (Tools Options Source Code Engineering <language>) 	<p>Synchronization Macros ^[1447]</p> <p>Language Options ^[1907]</p>

6.10.1.5.2 The Code Template Editor

The Code Template Editor provides the facilities of the *Common Code Editor*, including intellisense for the various macros. For more information on intellisense and the Common Code Editor, see the *Editing Source Code* topic.

Access **Settings | Code Generation Templates**

Reference

Option	Action	See also
Language	Select the programming language.	
New Language	Display the Programming Languages Datatypes dialog, which enables you to include programming languages other than those supported for Enterprise Architect, for which to create or edit code templates.	Programming Languages Datatypes ^[98]
Template	Display the contents of the active template, and provides the editor for modifying templates.	
Templates	List the base code templates; the active template is highlighted. The Modified field indicates whether you have changed the default template for the current language.	
Stereotype Overrides	List the stereotyped templates, for the active base template. The Modified field indicates whether you have modified a default stereotyped template.	
Add New Custom Template	Invoke a dialog for creating a custom stereotyped template.	
Add New Stereotyped Override	Invoke a dialog for adding a stereotyped template, for the currently selected base template.	
Get Default Template	Update the editor display with the default version of the active template.	
Save	Overwrite the active templates with the contents of the editor.	
Delete	If you have overridden the active template, the override is deleted and replaced by the corresponding default code template.	

Notes

- User-modified and user-defined Code Templates can be imported and exported as Reference Data (see the *Sharing Reference Data* topic); the templates defined for each language are indicated in the Export Reference Data dialog by the language name with the suffix *_Code_Templates* - if no templates

exist for a language, there is no entry for the language in the dialog

Learn more

- [The Code Template Editor in MDG Development](#)^[1462]
- [Sharing Reference Data](#)^[305]
- [Editing Source Code](#)^[1799]

6.10.1.5.3 Code Template Syntax

Code Templates are written using Enterprise Architect's Code Template Editor. The Code Template Editor supports syntax highlighting of the Code Template Framework language.

Topics

Topic	Detail	See also
Basic Constructs	Templates can contain: <ul style="list-style-type: none"> • Literal Text • Variables • Macros • Calls to other templates 	Literal Text ^[1406] Variables ^[1407] Macros ^[1409] Call Templates From Templates ^[1461]
Comments	If you want to add comments to the templates, use the command: <pre>\$COMMENT=" t e x t "</pre> where "text" is the text of the comment; this must be enclosed in quotes. The command is case-sensitive, so must be typed in upper case.	

Learn more

- [Code Template Framework](#)^[1394]
- [The Code Template Editor](#)^[1462]

6.10.1.5.3.1 Literal Text

All text within a given template that is not part of a macro or a variable definition/reference, is considered literal text. With the exception of blank lines, which are ignored, literal text is directly substituted from the template into the generated code.

Consider the following excerpt from the Java Class Declaration template:

```
$bases = "Base"
```

```
class %cl assName%$bases
```

On the final line, the word *class*, including the subsequent space, would be treated as literal text and thus for a Class named *foo* would return the output:

```
class fooBase
```

The blank line following the variable `$bases` would have no effect on the output.

Inserting System Characters:

The %, \$, " and \ characters have special meaning in the template syntax and cannot always be used as literal text. If these characters must be generated from within the templates, they can be safely reproduced using the following direct substitution macros:

Macro	Action
<code>%d!</code>	Produce a literal \$ character.
<code>%pc%</code>	Produce a literal % character.
<code>%qt%</code>	Produce a literal " character.
<code>%s!</code>	Produce a literal \ character

Notes

- String conjunction operators ("+" , "+=") are not required but can be used (see literals)

Learn more

- [Intermediary Language](#)^[17]

6.10.1.5.3.2 Variables

Template variables provide a convenient way of storing and retrieving data within a template. This section explains how variables are defined and referenced.

Variable Definitions:

Variable definitions take the basic form:

```
$<name> = <value>
```

where *<name>* can be any alpha-numeric sequence and *<value>* is derived from a macro or another variable.

A simple example definition would be:

```
$foo = %className%
```

Variables can be defined, using values from:

- Substitution, function or list macros
- String literals, enclosed within double quotation marks
- Variable references

Definition Rules:

The following rules apply to variable definitions:

- Variables have global scope within the template in which they are defined and are not accessible to other templates
- Each variable must be defined at the start of a line, without any intervening whitespace
- Variables are denoted by prefixing the name with \$, as in \$foo
- Variables do not have to be declared, prior to being defined
- Variables must be defined using either the assignment operator (=), or the addition-assignment operator (+=)
- Multiple terms can be combined in a single definition using the addition operator (+)

Examples

Using a substitution macro:

```
$foo = %opTag: " bar " %
```

Using a literal string:

```
$foo = " bar "
```

Using another variable:

```
$foo = $bar
```

Using a list macro:

```
$ops = %d i s t = " Oper a t i o n " @separ a t o r = " \ n \ n " @ n d e n t = " \ t " %
```

Using the addition-assignment operator (+=):

```
$body += %d i s t = " Oper a t i o n " @separ a t o r = " \ n \ n " @ n d e n t = " \ t " %
```

The above definition is equivalent to the following:

```
$body = $body + %d i s t = " Oper a t i o n " @separ a t o r = " \ n \ n " @ n d e n t = " \ t " %
```

Using multiple terms:

```
$templateArgs = %d i s t = " Cl a s s P a r a m e t e r " @separ a t o r = " , " %  
$template = "template<" + $templateArgs + ">"
```

Variable References:

Variable values can be retrieved by using a reference of the form:

`<name>`

where `<name>` can be a previously defined variable.

Variable references can be used in one of the following ways:

- As part of a macro, such as the argument to a function macro
- As a term in a variable definition
- As a direct substitution of the variable value into the output

It is legal to reference a variable before it is defined. In this case, the variable is assumed to contain an empty string value: ""

Example 1:

Using variables as part of a macro. The following is an excerpt from the default C++ *ClassNotes* template.

```

$wrapLen = %genOpt WrapComment % Define variables to store the style and wrap length
$style = % options.
genOpt CPPComment Style%

%if $style == "XML.NET" % Reference to $style as part of a condition.
%XML_COMMENT( $wrapLen) %
%else%
%CSTYLE_COMMENT( $wrapLen) % Reference to $wrapLen as an argument to function
%endif % macro.

```

Example 2:

Using variable references as part of a variable definitions.

```

foo = "foo" Define our variables.
bar = "bar"

foobar = foo + bar $foobar now contains the value foobar.

```

Example 3:

Substituting variable values into the output.

```

$bases=%classInherits% Store the result of the ClassInherits template in $bases.
Class %className%$bases Now output the value of $bases after the Class name.

```

6.10.1.5.3.3 Macros

Macros provide access to element fields within the UML model and are also used to structure the generated output. All macros are enclosed within percent (%) signs in the form of:

```
%<macroName>%
```

In general, macros (including the % delimiters) are substituted with literal text in the output. For example, consider the following item from the *Class Declaration* template:

```
... class %className% ...
```

The field substitution macro, `%className%`, would result in the current Class name being substituted in the output. So if the Class being generated was named `Foo`, the output would be:

```
... class Foo ...
```

The CTF contains a number of types of macros:

- [Template Substitution Macros](#) ^[1411]
- [Field Substitution Macros](#) ^[1411]
 - [Substitution Examples](#) ^[1413]
 - [Attribute Field Substitution Macros](#) ^[1414]
 - [Class Field Substitution Macros](#) ^[1416]
 - [Code Generation Option Field Substitution Macros](#) ^[1416]
 - [Connector Field Substitution Macros](#) ^[1423]
 - [Constraint Field Substitution Macros](#) ^[1426]
 - [Effort Field Substitution Macros](#) ^[1427]
 - [File Field Substitution Macros](#) ^[1427]
 - [File Import Field Substitution Macros](#) ^[1428]
 - [Link Field Substitution Macros](#) ^[1429]
 - [Linked File Field Substitution Macros](#) ^[1430]
 - [Metric Field Substitution Macros](#) ^[1431]
 - [Operation Field Substitution Macros](#) ^[1431]
 - [Package Field Substitution Macros](#) ^[1432]
 - [Parameter Field Substitution Macros](#) ^[1433]
 - [Problem Field Substitution Macros](#) ^[1434]
 - [Requirement Field Substitution Macros](#) ^[1435]
 - [Resource Field Substitution Macros](#) ^[1436]
 - [Risk Field Substitution Macros](#) ^[1436]
 - [Scenario Field Substitution Macros](#) ^[1437]
 - [Tagged Value Substitution Macros](#) ^[1437]
 - [Test Field Substitution Macros](#) ^[1438]
- [Function Macros](#) ^[1439]
- [Control Macros](#) ^[1444]
 - [List Macro](#) ^[1444]
 - [Branching Macros](#) ^[1446]
 - [Synchronization Macros](#) ^[1447]
 - [The PI Macro](#) ^[1448]
- [EASL Code Generation Macros](#) ^[1449]

Learn more

- [Intermediary Language](#)^[1714]

Template substitution macros correspond to [Base templates](#)^[1397]. These macros result in the execution of the named template. By convention, template macros are named according to Pascal casing.

Structure: %<TemplateName>%

where <TemplateName> can be one of the templates listed below.

When a template is referenced from within another template, it is generated with respect to the elements currently in scope. The specific template is selected based on the stereotypes of the elements in scope.

As noted previously, there is an implicit hierarchy among the various templates. Some care should be taken in order to preserve a sensible hierarchy of template references. For example, it does not make sense to use the %ClassInherits% macro within any of the attribute or operation templates. Conversely, the *Operation* and *Attribute* templates are designed for use within the *ClassBody* template.

The CTF contains the following template substitution macros:

- | | | |
|------------------------|------------------------------|----------------------------|
| • AttributeDeclaration | • ClassParameter | • NamespaceBody |
| • AttributeNotes | • File | • NamespaceDeclaration |
| • Attribute | • FileImpl | • NamespaceImpl |
| • Class | • ImportSection | • Operation |
| • ClassImpl | • ImportSectionImpl | • OperationBody |
| • ClassBase | • InnerClass | • OperationBodyImpl |
| • ClassBody | • InnerClassImpl | • OperationDeclaration |
| • ClassBodyImpl | • LinkedAttribute | • OperationDeclarationImpl |
| • ClassDeclaration | • LinkedAttributeNotes | • OperationImpl |
| • ClassDeclarationImpl | • LinkedAttributeDeclaration | • OperationNotes |
| • ClassInherits | • LinkedClassBase | • Parameter |
| • ClassInterface | • LinkedClassInterface | |
| • ClassNotes | • Namespace | |

The *field substitution* macros provide access to data in the model. In particular, they are used to access data fields from:

- Packages
- Classes
- Attributes
- Operations
- Parameters

Field substitution macros are named according to Camel casing. By convention, the macro is prefixed with an abbreviated form of the corresponding model element. For example, attribute-related macros begin with

att, as in the `%attName%` macro, to access the name of the attribute in scope.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

The following table lists a small number of project field substitution macros with a description of the result. See the *Learn more* section for tables listing groups of type-specific macros.

Macro Name	Description
eaDateTime	The current time with format: <i>DD-MMM-YYYY HH:MM:SS AM/PM</i> .
eaGUID	A unique GUID for this generation.
eaVersion	Program Version (located in an Enterprise Architect dialog by selecting Help About EA).

Learn more

- [Substitution Examples](#) ^[1413]
- [Attribute Field Substitution Macros](#) ^[1414]
- [Class Field Substitution Macros](#) ^[1416]
- [Code Generation Option Field Substitution Macros](#) ^[1418]
- [Connector Field Substitution Macros](#) ^[1423]
- [Constraint Field Substitution Macros](#) ^[1426]
- [Effort Field Substitution Macros](#) ^[1427]
- [File Field Substitution Macros](#) ^[1427]
- [File Import Field Substitution Macros](#) ^[1428]
- [Link Field Substitution Macros](#) ^[1429]
- [Linked File Field Substitution Macros](#) ^[1430]
- [Metric Field Substitution Macros](#) ^[1431]
- [Operation Field Substitution Macros](#) ^[1431]
- [Package Field Substitution Macros](#) ^[1432]
- [Parameter Field Substitution Macros](#) ^[1433]
- [Problem Field Substitution Macros](#) ^[1434]
- [Requirement Field Substitution Macros](#) ^[1435]
- [Resource Field Substitution Macros](#) ^[1436]
- [Risk Field Substitution Macros](#) ^[1436]
- [Scenario Field Substitution Macros](#) ^[1437]
- [Tagged Value Substitution Macros](#) ^[1437]
- [Test Field Substitution Macros](#) ^[1438]

Field substitution macros can be used in one of two ways:

- Direct Substitution or
- Conditional Substitution

Direct Substitution:

This form directly substitutes the corresponding value of the element in scope into the output.

Structure: %<macroName>%

Where <macroName> can be any of the macros listed in the Field Substitution Macros tables.

Examples

- %class Name %
- %op Name %
- %att Name %

Conditional Substitution

This form of the macro enables alternative substitutions to be made depending on the macro's value.

Structure: %<macroName> (== "<text>") ? <subTrue> (: <subFalse>) %

Where:

- (<text>) denotes that <text> is optional
- <text> is a string representing a possible value for the macro
- <subTrue> and <subFalse> can be a combination of quoted strings and the keyword value; where the value is used, it is replaced with the macro's value in the output.

Examples

- %classAbstract=="T" ? "pure" : "" %
- %opStereoType=="operator" ? "operator" : "" %
- %paramDefault!="" ? "= value" : "" %

The above three examples output nothing if the condition fails. In this case the false condition can be omitted, resulting in the following usage:.

- %classAbstract=="T" ? "pure" %
- %opStereoType=="operator" ? "operator" %

- `%paramDefault != "" ? " = " value%`

The third example of both blocks shows a comparison checking for a non-empty value or existence. This test can also be omitted.

- `%paramDefault ? " = " value : ""%`
- `%paramDefault ? " = " value%`

All of the above examples containing `paramDefault` are equivalent. If the parameter in scope had a default value of **10**, the output from each of them would normally be:

= 10

Notes

- In a conditional substitution macro, any white space following `<macroName>` is ignored; if white space is required in the output, it should be included within the quoted substitution strings

Learn more

- [Field Substitution Macros](#)^[141]

The following table lists each of the attribute field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
<code>attAlias</code>	Attributes dialog: Alias .
<code>attAllowDuplicates</code>	Attributes Detail dialog: Allow Duplicates checkbox.
<code>attClassifierGUID</code>	The unique GUID for the classifier of the current attribute.
<code>attCollection</code>	Attributes Detail dialog: Attribute is a Collection checkbox.
<code>attConst</code>	Attributes dialog: Const checkbox.
<code>attContainerType</code>	Attributes Detail dialog: Container Type .
<code>attContainment</code>	Attributes dialog: Containment .
<code>attDerived</code>	Attributes dialog: Derived checkbox.

Macro Name	Description
attGUID	The unique GUID for the current attribute.
attInitial	Attributes dialog: Initial .
attIsEnumLiteral	Attributes dialog: Is Literal checkbox.
attLength	Column dialog: Length .
attLowerBound	Attributes Detail dialog: Lower Bound .
attName	Attributes dialog: Name .
attNotes	Attributes dialog: Notes .
attOrderedMultiplicity	Attributes Detail dialog: Ordered Multiplicity checkbox.
attProperty	Attributes dialog: Property checkbox.
attQualType	The attribute type qualified by the namespace path (if generating namespaces) and the classifier path (dot delimited). If the attribute classifier has not been set, is equivalent to the <i>attType</i> macro.
attScope	Attributes dialog: Scope .
attStatic	Attributes dialog: Static checkbox.
attStereotype	Attributes dialog: Stereotype .
attType	Attributes dialog: Type .
attUpperBound	Attributes Detail dialog: Upper Bound .
attVolatile	Attributes Detail dialog: Transient checkbox.

Learn more

- [Field Substitution Macros](#)^[1417]

The following table lists each of the Class field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
elemType	The element type: Interface or Class.
classAbstract	Class dialog: Abstract checkbox.
classAlias	Class dialog: Alias .
classArguments	Class Detail dialog: C++ Templates: Arguments .
classAuthor	Class dialog: Author .
classBaseName	Type Hierarchy dialog: Class Name (for use where no connector exists between child and base Classes).
classBaseScope	The scope of the inheritance as reverse engineered. (For use where no connector exists between child and base Classes.)
classBaseVirtual	The virtual property of the inheritance as reverse engineered. (For use where no connector exists between child and base Classes.)
classComplexity	Class dialog: Complexity .
classCreated	The date and time the Class was created.
classGUID	The unique GUID for the current Class.
classHasConstructor	Looks at the list of methods in the current object and, depending on the conventions of the current language, returns T if one is a default constructor. Typically used with the genOptGenConstructor ^[1418] macro.

Macro Name	Description
classHasCopyConstructor	Looks at the list of methods in the current object and, depending on the conventions of the current language, returns T if one is a copy constructor. Typically used with the genOptGenCopyConstructor ^[1418] macro.
classHasDestructor	Looks at the list of methods in the current object and, depending on the conventions of the current language, returns T if one is a destructor. Typically used with the genOptGenDestructor ^[1418] macro.
classHasParent	True , if the Class in scope has one or more base Classes.
classImports	Code Gen dialog: Imports .
classIsActive	Class Advanced dialog: Is Active checkbox.
classIsInstantiated	True , if the Class is an instantiated template Class.
classIsLeaf	Class Advanced dialog: Is Leaf checkbox.
classIsRoot	Class Advanced dialog: Is Root checkbox.
classIsSpecification	Class Advanced dialog: Is Specification checkbox.
classKeywords	Class dialog: Keywords .
classLanguage	Class dialog: Language .
classMacros	A space separated list of macros defined for the Class.
classModified	The date and time the Class was last modified.
classMultiplicity	Class Advanced dialog: Multiplicity .
className	Class dialog: Name .
classNotes	Class dialog: Note .

Macro Name	Description
classParamDefault	Class Detail dialog.
classParamName	Class Detail dialog.
classParamType	Class Detail dialog.
classPersistence	Class dialog: Persistence .
classPhase	Class dialog: Phase .
classQualName	The Class name prefixed by its outer Classes. Class names are separated by double colons (::).
classScope	Class dialog: Scope .
classStereotype	Class dialog: Stereotype .
classStatus	Class dialog: Status .
classVersion	Class dialog: Version .

Learn more

- [Field Substitution Macros](#)^[1417]

The following table lists each of the code generation option field substitution macros with a description of the result.

These macros operate on the source code generation options defined in the Source Code Engineering pages of the Options dialog (**Tools | Options | Source Code Engineering**).

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
genOptActionScriptVersion	ActionScript Specifications page: Default Version .
genOptCDefaultAttributeType	C Specifications page: Default Attribute Type .

Macro Name	Description
genOptCGenMethodNotesInBody	C Specifications page: Method Notes In Implementation.
genOptCGenMethodNotesInHeader	C Specifications page: Method Notes In Header.
genOptCSynchNotes	C Specifications page: Synchronize Notes in Generation.
genOptCSynchCFile	C Specifications page: Synchronise Implementation file in Generation.
genOptCDefaultSourceDirectory	C Specifications page: Default Source Directory.
genOptCNamespaceDelimiter	C Specifications page: Namespace Delimiter.
genOptCOperationRefParam	C Specifications page: Reference as Operation Parameter.
genOptCOperationRefParamStyle	C Specifications page: Reference Parameter Style.
genOptCOperationRefParamName	C Specifications page: Reference Parameter Name.
genOptCConstructorName	C Specifications page: Default Constructor Name.
genOptCDestructorName	C Specifications page: Default Destructor Name.
genOptCPPCommentStyle	C++ Specifications page: Comment Style.
genOptCPPDefaultAttributeType	C++ Specifications page: Default Attribute Type.
genOptCPPDefaultReferenceType	C++ Specifications page: Default Reference Type.
genOptCPPDefaultSourceDirectory	C++ Specifications page: Default Source Directory.
genOptCPPGenMethodNotesInHeader	C++ Specifications page: Method Notes In Header checkbox.
genOptCPPGenMethodNotesInBody	C++ Specifications page: Method Notes In Body checkbox.

Macro Name	Description
dy	
genOptCPPGetPrefix	C++ Specifications page: Get Prefix.
genOptCPPHeaderExtension	C++ Specifications page: Header Extension.
genOptCPPSetPrefix	C++ Specifications page: Set Prefix.
genOptCPPSourceExtension	C++ Specifications page: Source Extension.
genOptCPPSynchCPPFile	C++ Specifications page: Synchronize Notes.
genOptCPPSynchNotes	C++ Specifications page: Synchronize CPP File.
genOptCSDefaultAttributeType	C# Specifications page: Default Attribute Type.
genOptCSSourceExtension	C# Specifications page: Default file extension.
genOptCSGenDispose	C# Specifications page: Generate Dispose.
genOptCSGenFinalizer	C# Specifications page: Generate Finalizer.
genOptCSGenNamespace	C# Specifications page: Generate Namespace.
genOptCSDefaultSourceDirectory	C# Specifications page: Default Source Directory.
genOptDefaultAssocAttName	Attribute/Operations Specifications page: Default name for associated.
genOptDefaultConstructorScope	Object Lifetimes page: Default Constructor Visibility.
genOptDefaultCopyConstructorScope	Object Lifetimes page: Default Copy Constructor Visibility.
genOptDefaultDatabase	Code Editors page: Default Database.
genOptDefaultDestructorScope	Object Lifetimes page: Default Destructor Constructor Visibility.

Macro Name	Description
genOptGenCapitalisedProperties	Source Code Engineering page: Capitalize Attribute Names for Properties checkbox.
genOptGenComments	Source Code Engineering page: Generate Comments checkbox.
genOptGenConstructor	Object Lifetimes page: Generate Constructor checkbox.
genOptGenConstructorInline	Object Lifetimes page: Constructor Inline checkbox.
genOptGenCopyConstructor	Object Lifetimes page: Generate Copy Constructor checkbox.
genOptGenCopyConstructorInline	Object Lifetimes page: Copy Constructor Inline checkbox.
genOptGenDestructor	Object Lifetimes page: Generate Destructor checkbox.
genOptGenDestructorInline	Object Lifetimes page: Destructor Inline checkbox.
genOptGenDestructorVirtual	Object Lifetimes page: Virtual Destructor checkbox.
genOptGenImplementedInterfaceOps	Attribute/Operations Specifications page: Generate methods for implemented interfaces checkbox.
genOptGenPrefixBoolProperties	Source Code Engineering page: Use is prefix for boolean property Get() .
genOptGenRoleNames	Source Code Engineering page: Autogenerate role names when creating code.
genOptGenUnspecAssocDir	Source Code Engineering page: Do not generate members where Association direction is unspecified checkbox.
genOptJavaDefaultAttributeType	Java Specifications page: Default attribute type.
genOptJavaGetPrefix	Java Specifications page: Get Prefix.

Macro Name	Description
genOptJavaDefaultSourceDirectory	Java Specifications page: Default Source Directory .
genOptJavaSetPrefix	Java Specifications page: Set Prefix .
genOptJavaSourceExtension	Java Specifications page: Source code extension .
genOptPHPDefaultSourceDirectory	PHP Specifications page: Default Source Directory .
genOptPHPGetPrefix	PHP Specifications page: Get Prefix .
genOptPHPSetPrefix	PHP Specifications page: Set Prefix .
genOptPHPSourceExtension	PHP Specifications page: Default file extension .
genOptPHPVersion	PHP Specifications page: PHP Version .
genOptPropertyPrefix	Source Code Engineering page: Remove prefixes when generating Get/Set properties .
genOptVBMultiUse	VB Specifications page: Multiuse checkbox.
genOptVBPersistable	VB Specifications page: Persistable checkbox.
genOptVBDataBindingBehavior	VB Specifications page: Data binding behavior checkbox.
genOptVBDataSourceBehavior	VB Specifications page: Data source behavior checkbox.
genOptVBGlobal	VB Specifications page: Global namespace checkbox.
genOptVBCreatable	VB Specifications page: Creatable checkbox.
genOptVBExposed	VB Specifications page: Exposed checkbox.
genOptVBMTS	VB Specifications page: MTS Transaction Mode .

Macro Name	Description
genOptVNetGenNamespace	VB.Net Specifications page: Generate Namespace .
genOptVBVersion	VB Specifications page: Default Version .
genOptWrapComment	Source Code Engineering page: Wrap length for comment lines .

Learn more

- [Field Substitution Macros](#) ^[141]

The following table lists each of the connector field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
connectorAlias	Connector Properties dialog: Alias .
connectorDestAccess	Connector Properties dialog, Target Role tab: Access .
connectorDestAggregation	Connector Properties dialog, Target Role tab: Aggregation .
connectorDestAlias	Connector Properties dialog, Target Role tab: Alias .
connectorDestAllowDuplicates	Connector Properties dialog, Target Role tab: Allow Duplicates checkbox.
connectorDestChangeable	Connector Properties dialog, Target Role tab: Changeable .
connectorDestConstraint	Connector Properties dialog, Target Role tab: Constraint(s) .
connectorDestContainment	Connector Properties dialog, Target Role tab: Containment .
connectorDestDerived	Connector Properties dialog, Target Role tab: Derived checkbox.
connectorDestDerivedUnion	Connector Properties dialog, Target Role tab: DerivedUnion checkbox.

Macro Name	Description
connectorDestElem*	A set of macros that access a property of the element at the target end of a connector. The * (asterisk) is a wildcard that corresponds to any class substitution macro in this list <i>connectorDestElemAlias (classAlias), connectorDestElemAuthor (classAuthor).</i>
connectorDestElemType	The element type of the connector destination element. (Separate from the <i>connectorDestElem*</i> macros because there is no <i>classType</i> substitution macro.)
connectorDestMemberType	Connector Properties dialog, Target Role tab: Member Type .
connectorDestMultiplicity	Connector Properties dialog, Target Role tab: Multiplicity .
connectorDestNavigability	Connector Properties dialog, Target Role tab: Navigability .
connectorDestNotes	Connector Properties dialog, Target Role tab: Role Notes .
connectorDestOrdered	Connector Properties dialog, Target Role tab: Ordered checkbox.
connectorDestOwned	Connector Properties dialog, Target Role tab: Owned checkbox.
connectorDestQualifier	Connector Properties dialog, Target Role tab: Qualifier(s) .
connectorDestRole	Connector Properties dialog, Target Role tab: Role .
connectorDestScope	Connector Properties dialog, Target Role tab: Target Scope .
connectorDestStereotype	Connector Properties dialog, Target Role tab: Stereotype .
connectorDirection	Connector Properties: Direction .
connectorEffect	Transition Constraints dialog: Effect .
connectorGuard	Object Flow and Transition Constraints dialog: Guard .

Macro Name	Description
connectorGUID	The unique GUID for the current connector.
connectorName	Connector Properties: Name .
connectorNotes	Connector Properties: Notes .
connectorSourceAccess	Connector Properties dialog, Source Role tab: Access .
connectorSourceAggregation	Connector Properties dialog, Source Role tab: Aggregation .
connectorSourceAlias	Connector Properties dialog, Source Role tab: Alias .
connectorSourceAllowDuplicates	Connector Properties dialog, Source Role tab: Allow Duplicates checkbox.
connectorSourceChangeable	Connector Properties dialog, Source Role tab: Changeable .
connectorSourceConstraint	Connector Properties dialog, Source Role tab: Constraint(s) .
connectorSourceContainment	Connector Properties dialog, Source Role tab: Containment .
connectorSourceDerived	Connector Properties dialog, Source Role tab: Derived checkbox.
connectorSourceDerivedUnion	Connector Properties dialog, Source Role tab: DerivedUnion checkbox.
connectorSourceElem*	A set of macros that access a property of the element at the source end of a connector. The * (asterisk) is a wildcard that corresponds to any class substitution macro in this list: <i>connectorSourceElemAlias</i> (<i>classAlias</i>), <i>connectorSourceElemAuthor</i> (<i>classAuthor</i>).
connectorSourceElemType	The element type of the connector source element. (Separate from the <i>connectorSourceElem*</i> macros because there is no <i>classType</i> substitution macro.)
connectorSourceMemberType	Connector Properties dialog, Source Role tab: Member Type .
connectorSourceMultiplicity	Connector Properties dialog, Source Role tab: Multiplicity .

Macro Name	Description
connectorSourceNavigability	Connector Properties dialog, Source Role tab: Navigability .
connectorSourceNotes	Connector Properties dialog, Source Role tab: Role Notes .
connectorSourceOrdered	Connector Properties dialog, Source Role tab: Ordered checkbox.
connectorSourceOwned	Connector Properties dialog, Source Role tab: Owned checkbox.
connectorSourceQualifier	Connector Properties dialog, Source Role tab: Qualifier(s) .
connectorSourceRole	Connector Properties dialog, Source Role tab: Role .
connectorSourceScope	Connector Properties dialog, Source Role tab: Target Scope .
connectorSourceStereotype	Connector Properties dialog, Source Role tab: Stereotype .
connectorStereotype	Connector Properties dialog: Stereotype .
connectorTrigger	Transition Constraints dialog: Trigger .
connectorType	The connector type; for example, Association or Generalization.
connectorWeight	Object Flow Constraints dialog: Weight .

Learn more

- [Field Substitution Macros](#)^[141]

The following table lists each of the Constraint field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
constraintName	Class dialog, Constraints tab: Name .

Macro Name	Description
constraintNotes	Class dialog, Constraints tab: Notes .
constraintStatus	Class dialog, Constraints tab: Status .
constraintType	Class dialog, Constraints tab: Type .
constraintWeight	Class dialog, Constraints tab: ordering (hand up/down) keys.

The following table lists each of the Effort field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
effortName	Project Management window: Effort .
effortNotes	Project Management window: Notes (unlabelled).
effortTime	Project Management window: Time .
effortType	Project Management window: Type .

The following table lists each of the Class field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
fileExtension	The file type extension of the file being generated.
fileName	The name of the file being generated.
fileNameImpl	The filename of the implementation file for this generation, if applicable.

Macro Name	Description
fileHeaders	Code Gen dialog: Headers .
fileImports	Code Gen dialog: Imports . For supported languages this also includes dependencies derived from associations.
filePath	The full path of the file being generated.
filePathImpl	The full path of the implementation file for this generation, if applicable.

The following table lists each of the file import field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
importClassName	The name of the Class being imported.
importFileName	The filename of the Class being imported.
importFilePath	The full path of the Class being imported.
importFromAggregation	T if the Class has an Aggregation connector to a Class in this file, F otherwise.
importFromAssociation	T if the Class has an Association connector to a Class in this file, F otherwise.
importFromAtt	T if an attribute of a Class in the current file is of the type of this Class, F otherwise.
importFromDependency	T if the Class has a Dependency connector to a Class in this file, F otherwise.
importFromGeneralization	T if the Class has a Generalization connector to a Class in this file, F otherwise.
importFromMeth	T if a method return type of a Class in the current file is the type of this Class, F otherwise.

Macro Name	Description
importFromParam	T if an method parameter of a Class in the current file is of the type of this Class, F otherwise.
importFromRealization	T if the Class has a Realization connector to a Class in this file, F otherwise.
importInFile	T if the Class is in the current file, F otherwise.
importPackagePath	The package path with a '.' separator of the Class being imported.
ImportRelativeFilePath	The relative file path of the Class being imported from the file path of the file being generated.

Learn more

- [Field Substitution Macros](#)^[141]

The following table lists each of the Link field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
linkAttAccess	Association Properties Target Role dialog: Access .
linkAttCollectionClass	The collection appropriate for the linked attribute in scope.
linkAttContainment	Association Properties Target Role dialog: Containment .
linkAttName	Association Properties dialog: Target .
linkAttNotes	Association Properties Target Role dialog: Role Notes .
linkAttQualName	The Association target qualified by the namespace path (if generating namespaces) and the classifier path (dot delimited).
linkAttRole	Association Properties Target Role dialog: Role .

Macro Name	Description
linkAttStereotype	Association Properties Target Role dialog: Stereotype .
linkAttTargetScope	Association Properties Target Role dialog: Target Scope .
linkCard	Link Properties Target Role dialog: Multiplicity .
linkGUID	The unique GUID for the current connector.
linkParentName	Generalization Properties dialog: Target .
linkParentQualName	The Generalization target qualified by the namespace path (if generating namespaces) and the classifier path (dot delimited).
linkStereotype	The stereotype of the current connector.
linkVirtualInheritance	Generalization Properties dialog: Virtual Inheritance .

The following table lists each of the Linked File field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
linkedFileLastWrite	Class Properties dialog: Last Write .
linkedFileNotes	Class Properties dialog: Notes .
linkedFilePath	Class Properties dialog: File Path .
linkedFileSize	Class Properties dialog: Size .
linkedFileType	Class Properties dialog: Type .

The following table lists each of the Metric field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
metricName	Project Management dialog, Metrics tab: Metric field.
metricNotes	Project Management dialog, Metrics tab: (Notes) field.
metricType	Project Management dialog, Metrics tab: Type field.
metricWeight	Project Management dialog, Metrics tab: Weight field.

The following table lists each of the Operation field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
opAbstract	Operation dialog: Virtual checkbox.
opAlias	Operation dialog: Alias .
opBehavior	Operation Behavior dialog: Behavior .
opCode	Operation Behavior dialog: Initial Code .
opConcurrency	Operation dialog: Concurrency .
opConst	Operation dialog: Const checkbox.
opGUID	The unique GUID for the current operation.
opImplMacros	A space-separated list of macros defined in the implementation of this operation.
opIsQuery	Operation dialog: IsQuery checkbox.

Macro Name	Description
opMacros	A space-separated list of macros defined in the declaration for this operation.
opName	Operation dialog: Name .
opNotes	Operation dialog: Notes .
opPure	Operation dialog: Pure checkbox.
opReturnArray	Operation dialog: Return Array checkbox.
opReturnClassifierGUID	The unique GUID for the classifier of the current operation.
opReturnQualType	The operation return type qualified by the namespace path (if generating namespaces) and the classifier path (dot delimited). If the return type classifier has not been set, is equivalent to the <i>opReturnType</i> macro.
opReturnType	Operation dialog: Return Type .
opScope	Operation dialog: Scope .
opStatic	Operation dialog: Static checkbox.
opStereotype	Operation dialog: Stereotype .
opSynchronized	Operation dialog: Synchronized checkbox.

Learn more

- [Field Substitution Macros](#)^[141]

The following table lists each of the Package field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
packageAbstract	Package dialog: Abstract .
packageAlias	Package dialog: Alias .
packageAuthor	Package dialog: Author .
packageComplexity	Package dialog: Complexity .
packageGUID	The unique GUID for the current package.
packageKeywords	Package dialog: Keywords .
packageLanguage	Package dialog: Language .
packageName	Package dialog: Name .
packagePath	The string representing the hierarchy of packages, for the Class in scope. Each package name is separated by a dot (.).
packagePhase	Package dialog: Phase .
packageScope	Package dialog: Scope .
packageStatus	Package dialog: Status .
packageStereotype	Package dialog: Stereotype .
packageVersion	Package dialog: Version .

Learn more

- [Field Substitution Macros](#)^[141]

The following table lists each of the Parameter field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
paramClassifierGUID	The unique GUID for the classifier of the current parameter.
paramDefault	Operation Parameters dialog: Default .
paramFixed	Operation Parameters dialog: Fixed checkbox.
paramGUID	The unique GUID for the current parameter.
paramIsEnum	True , if the parameter uses the <i>enum</i> keyword (C++).
paramKind	Operation Parameters dialog: Kind .
paramName	Operation Parameters dialog: Name .
paramNotes	Operation Parameters dialog: Notes .
paramQualType	The parameter type qualified by the namespace path (if generating namespaces) and the classifier path (dot delimited). If the parameter classifier has not been set, is equivalent to the <i>param Type</i> macro.
paramType	Operation Parameters dialog: Type .

The following table lists each of the Problem field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
problemCompletedBy	Maintenance dialog, Element Issues tab: Completed by .
problemCompletedDate	Maintenance dialog, Element Issues tab: Completed .
problemHistory	Maintenance dialog, Element Issues tab: History .
problemName	Maintenance dialog, Element Issues tab: Name .

Macro Name	Description
problemNotes	Maintenance dialog, Element Issues tab: Description .
problemPriority	Maintenance dialog, Element Issues tab: Priority .
problemRaisedBy	Maintenance dialog, Element Issues tab: Raised by .
problemRaisedDate	Maintenance dialog, Element Issues tab: Raised .
problemStatus	Maintenance dialog, Element Issues tab: Status .
problemVersion	Maintenance dialog, Element Issues tab: Version .

The following table lists each of the Requirement field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
requirementDifficulty	Properties dialog: Require tab: Difficulty .
requirementLastUpdated	Properties dialog: Require tab: Last Update .
requirementName	Properties dialog: Require tab: Short Description .
requirementNotes	Properties dialog: Require tab: Notes .
requirementPriority	Properties dialog: Require tab: Priority .
requirementStatus	Properties dialog: Require tab: Status .
requirementType	Properties dialog: Require tab: Type .

The following table lists each of the Resource field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
resourceAllocatedTime	Project Management window, Resource Allocation tab: Allocated Time .
resourceEndDate	Project Management window, Resource Allocation tab: End Date .
resourceExpectedTime	Project Management window, Resource Allocation tab: Expected Time .
resourceExpendedTime	Project Management window, Resource Allocation tab: Time Expended .
resourceHistory	Project Management window, Resource Allocation tab: History .
resourceName	Project Management window, Resource Allocation tab: Resource .
resourceNotes	Project Management window, Resource Allocation tab: Description .
resourcePercentCompleted	Project Management window, Resource Allocation tab: Completed(%) .
resourceRole	Project Management window, Resource Allocation tab: Role .
resourceStartDate	Project Management window, Resource Allocation tab: Start Date .

The following table lists each of the Risk field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
riskName	Project Management window, Risks tab: Risk .
riskNotes	Project Management window, Risks tab: (Notes).
riskType	Project Management window, Risks tab: Type .

Macro Name	Description
riskWeight	Project Management window, Risks tab: Weight .

The following table lists each of the Scenario field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
scenarioGUID	The unique ID for a scenario. Identifies the scenario unambiguously within a model.
scenarioName	Properties dialog, Scenario tab: Scenario .
scenarioNotes	Properties dialog, Scenario tab: (Notes).
scenarioType	Properties dialog, Scenario tab: Type .

Tagged Value macros are a special form of field substitution macros, which provide access to element tags and the corresponding Tagged Values. They can be used in one of two ways:

- Direct Substitution
- Conditional Substitution

Direct Substitution:

This form of the macro directly substitutes the value of the named tag into the output.

Structure: %<macroName>: " <tagName> " %

<macroName> can be one of:

- attTag
- classTag
- connectorDestElemTag
- connectorDestTag
- connectorSourceElemTag
- connectorSourceTag
- connectorTag

- linkAttTag
- linkTag
- opTag
- packageTag
- paramTag

This corresponds to the tags for attributes, Classes, operations, packages, parameters, connectors with both ends, elements at both ends of connectors and connectors including the attribute end.

`<tagName>` is a string representing the specific tag name.

Example

```
%opTag: " attribute" %
```

Conditional Substitution:

This form of the macro mimics the conditional substitution defined for field substitution macros.

```
Structure: %<macroName>: "<tagName>" ( == "<test>" ) ? <subTrue> ( : <subFalse> ) %
```

Where:

- `<macroName>` and `<tagName>` are as defined above
- (`<text>`) denotes that `<text>` is optional
- `<test>` is a string representing a possible value for the macro
- `<sub True>` and `<sub False>` can be a combination of quoted strings and the keyword value; where the value is used, it gets replaced with the macro's value in the output

Examples

```
%opTag: "oplnline" ? "inline" : "" %
%opTag: "oplnline" ? "inline" %
%classTag: "unsafe" == "true" ? "unsafe" : "" %
%classTag: "unsafe" == "true" ? "unsafe" %
```

Tagged Value macros use the same naming convention as field substitution macros.

Learn more

- [Field Substitution Macros](#)^[141]

The following table lists each of the Test field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
testAcceptanceCriteria	Testing window: Acceptance Criteria .
testCheckedBy	Testing window: Checked By .
testDateRun	Testing window: Last Run .
testClass	The Testing window tab (the type of test defined): Unit, Integration, System, Acceptance, Scenario.
testInput	Testing window: Input .
testName	Testing window: Test .
testNotes	Testing window: Description .
testResults	Testing window: Results .
testRunBy	Testing window: Run By .
testStatus	Testing window: Status .
testType	Testing window: Type .

Learn more

- [Field Substitution Macros](#) ^[141]

Function macros are a convenient way of manipulating and formatting various element data. Each function macro returns a result string. There are two primary ways to use the results of function macros:

- Direct substitution of the returned string into the output, such as: `%TO_LOWER(attName)%`
- Storing the returned string as part of a variable definition such as: `$name = %TO_LOWER(attName)%`

Function macros can take parameters, which can be passed to the macros as:

- String literals, enclosed within double quotation marks
- Direct substitution macros without the enclosing percent signs
- Variable references

- Numeric literals

Multiple parameters are passed using a comma-separated list.

Function macros are named according to the All-Caps style, as in: **%CONVERT_SCOPE(opScope)%**

The available function macros are described below. Parameters are denoted by angle brackets, as in: *FUNCTION_NAME(<param>)*.

CONVERT_SCOPE(<umlScope>):

For use with supported languages, to convert <umlScope> to the appropriate scope keyword for the language being generated. The following table shows the conversion of <umlScope> with respect to the given language.

Language	Package	Public	Private	Protected
C++	public	public	private	protected
C#	internal	public	private	protected
Delphi	protected	public	private	protected
Java		public	private	protected
PHP	public	public	private	protected
VB	Protected	Public	Private	Protected
VB .Net	Friend	Public	Private	Protected

COLLECTION_CLASS(<language>):

Gives the appropriate collection Class for the language specified for the current linked attribute.

CSTYLE_COMMENT(<wrap_length>):

Converts the notes for the element currently in scope to plain C-style comments, using /* and */.

DELPHI_PROPERTIES(<scope>, <separator>, <indent>):

Generates a Delphi property.

DELPHI COMMENT(<wrap length>):

Converts the notes for the element currently in scope to Delphi comments.

EXEC_ADD_IN(<addin name>, <function name>, <prm 1>, ..., <prm n>):

Invokes an Enterprise Architect Add-In function, which can return a result string.

<addin_name> and <function_name> specify the names of the Add-In and function to be invoked.

Parameters to the Add-In function can be specified via parameters <prm_1> to <prm_n>.

```
$result = %EXEC_ADD_IN(" MyAddIn", " ProcessOperation", classGUID, opGUID) %
```

Any function that is to be called by the *EXEC_ADD_IN* macro must have two parameters: an *EA.Repository* object, and a *Variant* array that contains any additional parameters from the *EXEC_ADD_IN* call. Return type should be *Variant*.

```
Public Function ProcessOperation( Repository As EA.Repository, args As Variant ) As Variant
```

FIND(<src>, <subString>):

Position of the first instance of <subString> in <src>; -1 if none.

GET_ALIGNMENT():

Returns a string where all of the text on the current line of output is converted into spaces and tabs.

JAVADOC_COMMENT(<wrap length>):

Converts the notes for the element currently in scope to *javadoc*-style comments.

LEFT(<src>, <count>):

The first <count> characters of <src>.

LENGTH(<src>):

Length of <src>. Returns a string.

MID(<src>, <count>)**MID(<src>, <start>, <count>):**

Substring of <src> starting at <start> and including <count> characters. Where <count> is omitted the rest of the string is included.

PI(<option>, <value>, {<option>, <value>}):

Sets the PI for the current template to <value>. Valid <values> are:

- "\n"
- "\t "
- " " "
- " " "

<option> controls when the new PI takes effect. Valid values for <option> are:

- *I, Immediate*: the new PI is generated before the next non-empty template line
- *N, Next*: the new PI is generated after the next non-empty template line

Multiple pairs of options are allowed in one call. An example of the situation where this would be used is where one keyword is always on a new line, as illustrated below:

```
%PI = " " %
%classAbstract ? "abstract" %
%i f classTag: "macro" != " " %
%PI ( "I", "\n", "N", " " ) %
%classTag: "macro" %
%endif %
class
%class Name %
```

For more details, see the [description of PI](#)^[144B].

PROCESS END OBJECT(<template name>):

Enables the Classes that are one Class further away from the base Class, to be transformed into objects (such as attributes, operations, packages, parameters and columns) of the base Class. <template_name> refers to the working template that temporarily stores the data.

REMOVE DUPLICATES(<source>, <separator>):

Where <source> is a <separator> separated list; this removes any duplicate or empty strings.

REPLACE(<string>, <old>, <new>):

Replaces all occurrences of <old> with <new> in the given string <string>.

RESOLVE OP NAME():

Resolves clashes in interface names where two method-from interfaces have the same name.

RESOLVE QUALIFIED TYPE()

RESOLVE QUALIFIED TYPE(<separator>)

RESOLVE QUALIFIED TYPE(<separator>, <default>):

Generates a qualified type for the current attribute, linked attribute, linked parent, operation, or parameter. Enables the specification of a separator other than . and a default value for when some value is required.

RIGHT(<src>, <count>):

The last <count> characters of <src>.

TO LOWER(<string>):

Converts <string> to lower case.

TO UPPER(<string>):

Converts <string> to upper case.

TRIM(<string>)**TRIM(<string>, <trimChars>):**

Removes trailing and leading white spaces from <string>. If <trimChars> is specified, all leading and trailing characters in the set of <trimChars> are removed.

TRIM LEFT(<string>)**TRIM LEFT(<string>, <trimChars>):**

Removes the specified leading characters from <string>.

TRIM RIGHT(<string>)**TRIM RIGHT(<string>, <trimChars>):**

Removes the specified trailing characters from <string>.

VB COMMENT(<wrap length>):

Converts the notes for the element currently in scope to Visual Basic style comments.

WRAP COMMENT(<comment>, <wrap length>, <indent>, <start string>):

Wraps the text <comment> at width <wrap_length> putting <indent> and <start_string> at the beginning of each line.

```
$behavior = %WRAP_COMMENT(opBehavior, "40", " ", "//") %
```

<wrap_length> must still be passed as a string, even though **WRAP_COMMENT** treats this parameter as an integer.

WRAP LINES(<text>, <wrap length>, <start string>(, <end string>):

Wraps <text> as designated to be <wrap_length>, adding <start_string> to the beginning of every line and <end_string> to the end of the line if it is specified.

XML COMMENT(<wrap length>):

Converts the notes for the element currently in scope to XML-style comments.

Control macros are used to control the processing and formatting of the templates. The basic types of control macro include:

- The *list* macro, for generating multiple element features, such as attributes and operations
- The branching macros, which form *if-then-else* constructs to conditionally execute parts of a template
- The PI macro for formatting new lines in the output, which takes effect from the next non-empty line
- A PI function macro that enables setting PI to a variable and adds the ability to set the PI that is generated before the next line
- The synchronization macros

In general, control macros are named according to Camel casing.

Learn more

- [List Macro](#)^[1444]
- [Function Macros](#)^[1439]
- [Branching Macros](#)^[1446]
- [Synchronization Macros](#)^[1447]
- [The PI Macro](#)^[1448]

The *%list* macro performs an iterative pass on all the objects in the scope of the current template, and calls a template to process each of these.

The basic structure is:

```
%list =<TemplateName> @separator=<string> @indent=<string> ( <conditions>
) %
```

where *<string>* is a double-quoted literal string and *<TemplateName>* can be one of the following template names:

- Attribute
- Class
- ClassBase
- ClassImpl
- ClassInterface
- Constraint
- Custom Template (custom templates enable you to define your own templates)
- Effort
- InnerClass
- InnerClassImpl
- LinkedFile
- Metric
- Namespace

- Operation
- OperationImpl
- Parameter
- Problem
- Requirement
- Resource
- Risk
- Scenario
- Test

<conditions> is optional and appears the same as the conditions for *if* and *elseif* statements.

Example

The List command is the method to loop or iterate through a set of Objects that are contained or are under the current object.

For instance, in a Class transform, the Class might contain multiple Attributes; the following example calls the Attribute transform and outputs the result of processing the transform for each attribute of the Class in scope. The resultant list separates its items with a single new line and indents them two spaces respectively. If the Class in scope had any *stereotyped* attributes, they would be generated using the appropriately specialized template.

```
%list = "Attribute" @separator = "\n" @indent = "  "%
```

The *separator* attribute, denoted above by *@separator*, specifies the space that should be used between the list items, excluding the last item in the list.

The *indent* attribute, denoted by *@indent*, specifies the space by which each line in the generated output should be indented.

Special Cases:

There are some special cases to consider when using the *%list* macro:

- If the *Attribute* template is used as an argument to the *%list* macro, this also generates attributes derived from associations by executing the appropriate *LinkedAttribute* template
- If the *ClassBase* template is used as an argument to the *%list* macro, this also generates Class bases derived from links in the model by executing the appropriate *LinkedClassBase* template
- If the *ClassInterface* template is used as an argument to the *%list* macro, this also generates Class bases derived from links in the model by executing the appropriate *LinkedClassInterface* template
- If *InnerClass* or *InnerClassImpl* is used as an argument to the *%list* macro, these Classes are generated using the *Class* and *ClassImpl* templates respectively; these arguments tell Enterprise Architect that it should process the templates based on the inner Classes of the Class in scope

Learn more

- [Custom Templates](#)^[1462]
- [Control Macros](#)^[1444]

Branching macros provide if-then-else constructs. The CTF supports a limited form of branching through the following macros:

- *if*
- *elseif*
- *else*
- *endif*
- *endTemplate* (which exits the current template)

The basic structure of the *if* and *elseif* macros is:

```
%i f <test> <operator> <test>%
```

where *<operator>* can be one of:

- ==
- !=

and *<test>* can be one of:

- a string literal, enclosed within double quotation marks
- a direct substitution macro, without the enclosing percent signs
- a variable reference

Branches can be nested, and multiple conditions can be specified using one of:

- *and or*
- *or*

When specifying multiple conditions, *and* and *or* have the same order of precedence, and conditions are processed left to right.

If conditional statements on strings are case sensitive, "a *String*" does not equal "A *STRING*". Hence in some situations it is better to set the variable *\$str= TO_LOWER(variable)* or *TO_UPPER(variable)* then compare to a specific case.

Macros are not supported in the conditional statements. It is best to assign the results of a macro (string) to a variable, and then use the variable in the comparison.

```
$fldType = %TO_LOWER($parameter1)%  
$COMMENT = "Use the first 4 characters for Date and Time field types"  
$fldType4 = %LEFT($fldType, 4)%  
%i f $fldType4=="date"%  
Datetime  
%endif%
```

This takes a parameter of value "Datetime", "DATETIME" or "Date", and returns "Datetime".

The *endif* or *endTemplate* macros must be used to signify the end of a branch. In addition, the *endTemplate* macro causes the template to return immediately, if the corresponding branch is being executed.

Example 1:

```
%i f e l e m T y p e == " I n t e r f a c e " %
:
%e l s e %
%O p e r a t i o n B o d y %
%e n d I f %
```

In this case:

- If the *elemType* is "Interface" a semi-colon is returned
- If the *elemType* is not "Interface", a template called Operation Body is called

Example 2:

```
$b a s e s = " C l a s s B a s e "
$ i n t e r f a c e s = " " %
%i f $ b a s e s ! = " " a n d $ i n t e r f a c e s ! = " " %
: $ b a s e s , $ i n t e r f a c e s
%e l s e l f $ b a s e s ! = " " %
: $ b a s e s
%e l s e l f $ i n t e r f a c e s ! = " " %
: $ i n t e r f a c e s
%e n d I f %
```

In this case the text returned is : C l a s s B a s e .

Conditions using Boolean Value:

When doing branching using conditions that involve a system checkbox (boolean fields), such as Attribute.Static (*attStatic*) the conditional statement would be written as:

```
%i f a t t S t a t i c == " T " %
```

For example:

```
%i f a t t C o l l e c t i o n == " T " o r a t t O r d e r e d M u l t i p l i c i t y == " T " %
%e n d T e m p l a t e %
```

Learn more

- [Attribute Field Substitution Macros](#) ^[1415]

The *synchronization macros* are used to provide formatting hints to Enterprise Architect when inserting new sections into the source code, during forward synchronization. The values for synchronization macros must be set in the **File** templates.

The structure for setting synchronization macros is:

```
%<name>=<value>%
```

where *<name>* can be one of the macros listed below and *<value>* is a literal string enclosed by double quotes.

Macro Name	Description
synchNewClassNotesSpace	Space to append to a new Class note. Default value: \ n .
synchNewAttributeNotesSpace	Space to append to a new attribute note. Default value: \ n .
synchNewOperationNotesSpace	Space to append to a new operation note. Default value: \ n .
synchNewOperationBodySpace	Space to append to a new operation body. Default value: \ n .
synchNamespaceBodyIndent	Indent applied to Classes within non-global namespaces. Default value: \ t .

The PI (Processing Instruction) macro provides a means of defining the separator text to be inserted between the code pieces (which represent entities) that are generated using a template.

The structure for setting the Processing Instruction is:

```
%PI =<value>%
```

where <value> is a literal string enclosed by double quotes, with the following options:

- " \ n " - New line (the default)
- " " - Space
- " \ t " - Tab
- " " - Null

By default, the PI is set to generate a new line (\ n) for each non-empty substitution, which behavior can be changed by resetting the *PI* macro. For instance, a Class's Attribute declaration in simple VB code would be generated to a single line statement (with no new lines). These properties are derived from the Class-Attribute Properties in the model to generate, for example:

```
Private Const PrintFormat As String = "Portrait"
```

The template for generating this starts with the PI being set to a space rather than a new line:

```
%PI=" "%
%CONVERT_SCOPE(attScope)%
%endIf%
%i f attConst == "T"%
Const
%endIf%
```

On transforming this, *attScope* returns the VB keyword "Private" and *attConst* returns "Const" on the same line spaced by a single space (fitting the VB Class.Attribute definition example above).

Alternatively, when generating a Class you might want the Class declaration, the notes and Class body all separated by double lines. In this case the %PI is set to "/ n/ n" to return double line spacing:

```
%PI = "\n\n"%
%ClassDeclaration%
%ClassNotes%
%ClassBody%
```

PI Characteristics:

- Blank lines have no effect on the output
- Any line that has a macro that produces an empty result does not result in a PI separator (space/new line)
- The last entry does not return a PI; for example, %Classbody% (above) does not have a double line added after the body

Learn more

- [Function Macros](#)^[1447]

Enterprise Architect provides two Enterprise Architect Simulation Library (EASL) code generation macros to generate code from behavioral models. These are:

- *EASL_GET* and
- *EASLList*

EASL_GET:

The *EASL_GET* macro is used to retrieve a property or a collection of an EASL object. The EASL objects and the properties and collections for each object are identified in the [EASL Collections](#)^[1451] and [EASL Properties](#)^[1454] topics.

Syntax:

```
$result = %EASL_GET( <<Property>>, <<Owner ID>>, <<Name>>)
```

where:

- <<Property>> is either "Property" or "Collection"
- <<OwnerID>> is the ID of the owner object for which the property/collection is to be retrieved
- <<Name>> is the name of the property or Collection being accessed
- \$result is the returned value; this is "" if not a valid property

Example

```
$sPropName = %EASL_GET( " Property", $context, " Name" ) %
```

EASLList:

The *EASLList* macro is used to render each object in an EASL collection using the appropriate template.

Syntax:

```
$result = %EASLList=<<TemplateName>> @separator=<<Separator>>
          @indent=<<indent>> @owner=<<OwnedID>>
          @collection=<<CollectionName>> @option1=<<OPTION1>>
          @option2=<<OPTION2>>..... @optionN=<<OPTIONN>>%
```

where:

- <<TemplateName>> is the name of any [behavioral model template](#)^[1450] or [custom template](#)^[1462]
- <<Separator>> is a list separator (such as “\n”)
- <<indent>> is any indentation to be applied to the result
- <<owner>> is the ID of the object that contains the required collection
- <<CollectionName>> is the name of the required collection
- <<OPTION1>>....<<OPTION99>> are miscellaneous options that might be passed on the template; each option is given as an additional input parameter to the template
- \$result is the resultant value; this is “” if not a valid collection

Example

```
$sStates = %EASLList="State" @separator="\n" @indent="\t"
          @owner=$StateMachineGUI D @collection="States"
          @option=$sOption%
```

Behavioral Model Templates:

- Action
- Action Assignment
- Action Break
- Action Call
- Action Create
- Action Destroy
- Action If
- Action Loop
- Action Opaque
- Action Parallel
- Action RaiseEvent
- Action RaiseException
- Action Switch
- Behavior
- Behavior Body

- Behavior Declaration
- Behavior Parameter
- Call Argument
- Guard
- Property Object
- Property Declaration
- Property Notes
- State
- State CallBack
- State Enumerate
- State EnumeratedName
- StateMachine
- StateMachine HistoryVar
- Transition
- Transition Effect
- Trigger

This topic lists the EASL collections for each of the EASL objects, as retrieved by the [EASL_GET](#)^[1449] code generation macro.

Action:

Collection Name	Description
Arguments	The Action's arguments.
SubActions	The sub-actions of the Action.

Behavior:

Collection Name	Description
Actions	The Behavior's Actions.
Nodes	The Behavior's nodes.
Parameters	The Behavior's parameters.
Variables	The Behavior's variables.

Classifier:

Collection Name	Description
AllStateMachines	All State Machines for the Classifier.
AsynchProperties	The asynchronous properties of the Classifier.
AsynchTriggers	The asynchronous triggers of the Classifier.
Behaviors	The behaviors of the Classifier.
Properties	The properties of the Classifier.
TimedProperties	The timed properties of the Classifier.
TimedTriggers	The timed triggers of the Classifier.
Triggers	All triggers of the Classifier.

Construct:

Collection Name	Description
AllChildren	The Construct's children.
ClientDependencies	The client dependencies on the Construct.
StereoTypes	The stereotypes of the Construct.
SupplierDependencies	The supplier dependencies on the Construct.

Node:

Collection Name	Description
IncomingEdges	The Node's incoming edges.

Collection Name	Description
OutgoingEdges	The Node's outgoing edges.
SubNodes	The sub-nodes of the Node.

State:

Collection Name	Description
DoBehaviors	The State's Do behaviors.
EntryBehaviors	The State's Entry behaviors.
ExitBehaviors	The State's Exit behaviors.

StateMachine:

Collection Name	Description
AllFinalStates	The State Machine's final States.
AllStates	All States within the State Machine, including those within Submachine States.
DerivedTransitions	The State Machine's derived transitions with the associated valid effect.
States	The States within the State Machine.
Transitions	The transitions within the State Machine.
Vertices	The State Machine's vertices.

Transition:

Collection Name	Description
Effects	The Transition's effects.
Guards	The Transition's guards.
Triggers	The Transition's triggers.

Trigger:

Collection Name	Description
TriggeredTransitions	The triggered transitions associated with the Trigger.

Vertex:

Collection Name	Description
DerivedOutgoingTransitions	The Vertex's derived outgoing transitions after traversing the pseudo-nodes.
IncomingTransitions	The Vertex's incoming transitions.
OutgoingTransitions	The Vertex's outgoing transitions.

This topic lists the EASL properties for each of the EASL objects, as retrieved by the [EASL GET](#)^[1449] code generation macro.

Action:

Property Name	Description
Behavior	The Action's associated behavior (<i>Call Behavior Action</i> or <i>Call Operation Action</i>).
Body	The Action's body.
Context	The Action's context.
Guard	The Action's guard.

Property Name	Description
IsFinal	A check on whether the action is a final Action.
IsGuarded	A check on whether the action is a guarded Action.
IsInitial	A check on whether the action is an initial Action.
Kind	The Action's kind.
Next	The Action's next action.
Node	The Action's associated node in the graph.

Argument:

Property Name	Description
Parameter	The ID of the Argument's associated parameter.
Value	The default value of the argument.

Behavior:

Property Name	Description
InitialAction	The Behavior's initial action.
isReadOnly	The isReadOnly of the Behavior.
isSingleExecution	The isSingleExecution of the Behavior.
Kind	The kind of Behavior.
ReturnType	The return type of the Behavior.

Property Name	Description
Specification	The specification of the Behavior.

CallEvent:

Property Name	Description
Operation	The operation of the CallEvent.

ChangeEvent:

Property Name	Description
ChangeExpression	The change expression of the ChangeEvent.

Classifier:

Property Name	Description
HasBehaviors	A check on whether the Classifier has behavioral models (Activity and Interaction).
Language	The Classifier's language.
StateMachine	The State Machine of the Classifier.

Condition:

Property Name	Description
Expression	The Condition's expression.
Lower	The Condition's lower value.
Upper	The Condition's upper value.

Construct:

Property Name	Description
GetTaggedValue	The Property's Tagged Value.
IsStereotypeApplied	A check on whether a particular stereotype is applied to the Property.
Notes	Notes on the Property.
UMLType	The UML type of the Property.
Visibility	The visibility of the Property.

Edge:

Property Name	Description
From	The ID of the node from which the Edge arises.
To	The ID of the node at which the Edge is targeted.

EventObject:

Property Name	Description
EventKind	The event kind of the Event Object.

Instance:

Property Name	Description
Classifier	The classifier of the Instance.
Value	The value of the Instance.

Parameter:

Property Name	Description
Direction	The direction of the Parameter.
Type	The type of the Parameter.
Value	The value of the parameter.

Primitive:

Property Name	Description
FQName	The FQ name of the Primitive.
ID	The ID of the Primitive.
Name	The name of the Primitive.
ObjectType	The object type of the Primitive.
Parent	The IDParent of the Primitive.

PropertyObject:

Property Name	Description
BoundSize	The bound size of the PropertyObject (if it is a collection).
ClassifierStereoType	The stereotype of the PropertyObject's classifier.
IsAsynchProp	A check on whether the PropertyObject is an asynchronous property.
IsCollection	A check on whether the PropertyObject is a collection.
IsOrdered	A check on whether the PropertyObject is ordered (if it is a collection).

Property Name	Description
IsTimedProp	A check on whether the PropertyObject is a timed property.
Kind	The PropertyObject's kind.
LowerValue	The PropertyObject's lower value (if it is a collection).
Type	The PropertyObject's type.
UpperValue	The PropertyObject's upper value (if it is a collection).
Value	The PropertyObject's value.

SignalEvent:

Property Name	Description
Signal	The signal of the SignalEvent.

State:

Property Name	Description
HasSubMachine	A check on whether the State is a Submachine state.
IsFinalState	A check on whether the State is a final state.
SubMachine	Get the ID of the Submachine contained by the State (if applicable).

StateMachine:

Property Name	Description
HasSubMachineState	A check on whether the State Machine has a Submachine state.

Property Name	Description
InitialState	The State Machine's initial state.
SubMachineState	The State Machine's Submachine state.

TimeEvent:

Property Name	Description
When	The 'when' property of the TimeEvent.

Transition:

Property Name	Description
HasEffect	A check on whether the transition has a valid effect.
IsDerived	A check on whether the transition is a derived transition.
IsTranscend	A check on whether the transition transcends from one State Machine (Submachine state) to another.
IsTriggered	A check on whether the transition is triggered.
Source	The Transition's source.
Target	The Transition's target.

Trigger:

Property Name	Description
AsynchDestinationState	The asynchronous destination state of the Trigger (if it is an asynchronous trigger).
DependentProperty	The ID of the property associated with the Trigger.

Property Name	Description
Event	The Trigger's event.
Name	The Trigger's name.
Type	The Trigger's type.

Vertex:

Property Name	Description
IsHistory	A check on whether the vertex is a history state.
IsPseudoState	A check on whether the vertex is a pseudo state.
PseudoStateKind	The Vertex's pseudo-state kind.

6.10.1.5.3.4 Call Templates From Templates

Using function calls with parameters, you can call templates from other templates, whether standard templates or user-defined templates created within your project. Also, called templates can return a value, and can be called recursively.

Examples

A call statement returning a parameter to a variable:

```
$Source = %StateEnumeratedName($Source)%
```

A call statement to a template that has parameters:

```
%RuleTask($GUID, $index)%
```

Using the \$parameter statement in the called template:

```
$GUID = $parameter1
$index = $parameter2
```

Templates support recursive calls, such as the following recursive call on the template RuleTask:

```
$GUID = $parameter1
$index = $parameter2

%PI = ""%
```

```
$nul = "Initialize condition and action object"

$count = %BR_GET("RuleCount")%

%if $count==" " or $count == $index%
%ComputeRule($GUID)%
\n
%endTemplate%

%Rule($index)%
\n
$index = %MATH_ADD($index, "1")%
%RuleTask($GUID, $index)%
```

Learn more

- [Custom Templates](#)^[1462]
- [The Code Template Editor in MDG Development](#)^[1462]

6.10.1.5.4 The Code Template Editor in MDG Development

The following topics describe how you use the Code Template Editor window to create custom templates:

- [Custom Templates](#)^[1462]
- [Override Default Templates](#)^[1464]
- [Add New Stereotyped Templates](#)^[1464]
- [Create Templates For Custom Languages](#)^[1466]

The Code Template Editor provides the facilities of the **Common Code Editor**, including intellisense for the code generation template macros. For more information on intellisense and the Common Code Editor, see the *Editing Source Code* topic.

Learn more

- [Code Template Framework](#)^[1394]
- [Code Generation Template Macros](#)^[1409]
- [Editing Source Code](#)^[1799]

6.10.1.5.4.1 Custom Templates

Custom templates enable you to generate an element in many different ways. Enterprise Architect enables you to define custom templates that are associated with given elements and call these templates from existing templates. You can even add stereotype overrides to your custom templates. For example, you might list all of your parameters and their notes in your method notes.

Access **Settings | Code Generation Templates (Ctrl+Shift+P)**

How to

To create a new custom template using the Code Templates Editor

Step	Description	See also
1	In the Language field, click on the drop-down arrow and select the appropriate language.	
2	Click on the Add New Custom Template button. The Create New Custom Template dialog displays.	
3	<p>In the Template Type field, click on the drop-down arrow and select the appropriate element.</p> <p>The elements currently supported are:</p> <ul style="list-style-type: none"> • Attribute • Class • Class Base • Class Interface • Class Parameter • Connector • Import • Linked Attribute • Linked Class Base • Linked Class Interface • Namespace • Operation • Parameter <p><None> requires special treatment; it enables the definition of a function macro that doesn't actually apply to any of the types, but must be called as a function to define variables <i>\$parameter1</i>, <i>\$parameter2</i> and so on for each value passed in.</p>	
4	In the Template Name field, type an appropriate name. Click on the OK button.	
5	On the Code Templates Editor tab, the new template displays in the Templates list with the value Yes in the Modified field. The template is called <i><Template Type>_<Template Name></i> .	
6	Select the appropriate template from the Templates list and edit the contents in the Template field to meet your requirements.	
7	Click on the Save button.	

	This stores the new stereotyped template in the .EAP file. The template is now available from the list of templates and via direct substitution for use.	
--	--	--

6.10.1.5.4.2 Override Default Templates

Enterprise Architect has a set of built-in or default code generation templates. The Code Templates Editor enables you to modify these default templates, hence customizing the way in which Enterprise Architect generates code. You can choose to modify any or all of the base templates to achieve your required coding style.

Any templates that you have overridden are stored in the .EAP file. When generating code, Enterprise Architect first checks whether a template has been modified and if so, uses that template. Otherwise the appropriate default template is used.

Access **Configuration | Code Generation Templates**

Reference

Override a default code generation template using the Code Templates Editor, as indicated below:

Field/Button	Description	See also
Language	Select the appropriate language from the list.	
Templates	Select one of the base templates from the list.	
Stereotype Overrides	If the base template has stereotyped overrides, you can select one of these from the list.	
<Other fields>	Make any other modifications required.	
Save	Click on this button to store the modified version of the template to the .EAP file. The template is marked as modified.	

When generating code, Enterprise Architect now uses the overridden template, instead of the default template.

6.10.1.5.4.3 Add New Stereotyped Templates

Sometimes it is useful to define a specific code generation template for use with elements of a given stereotype. This enables different code to be generated for elements, depending on their stereotype. Enterprise Architect provides some default templates, which have been specialized for commonly used stereotypes in supported languages. For example the Operation Body template for C# has been specialized for the *property* stereotype, so that it automatically generates its constituent *get* and *set* methods. Users can override the default stereotyped templates as described in the *Override Default Templates* topic. Additionally users can define templates for their own stereotypes, as described below.

Access [Configuration](#) | [Code Generation Templates](#)

How to

To add a new stereotyped template using the Code Templates Editor

Step	Description	See also
1	Select the appropriate language, from the Language list.	
2	Select one of the base templates, from the Templates list.	
3	Click on the Add New Stereotyped Override button. The New Template Override dialog displays.	
4	Select the required Feature and/or Class stereotype. Click on the OK button.	
5	The new stereotyped template override displays in Stereotype Overrides list, marked as modified.	
6	Make the required modifications in the Code Templates Editor.	
7	Click on the Save button to store the new stereotyped template in the .EAP file. Enterprise Architect can now use the stereotyped template, when generating code for elements of that stereotype.	

Notes

- Class and feature stereotypes can be combined to provide a further level of specialization for features; for example, if properties should be generated differently when the Class has a stereotype *MyStereotype*, then both *property* and *MyStereotype* should be specified in the New Template Override dialog

Learn more

- [Override Default Templates](#) ^[1465]

6.10.1.5.4.4 Create Custom Language Template

Enterprise Architect can forward generate code for languages that it does not specifically support, if the appropriate code generation templates are defined for that language. This topic outlines the steps required to define templates for custom languages.

How to

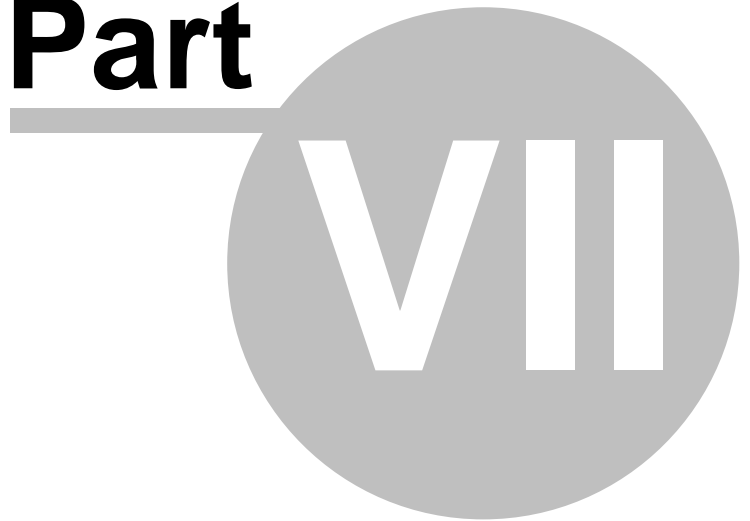
To define a template for a custom language

Step	Description	See also
1	<p>Create the custom language as a new product:</p> <ul style="list-style-type: none"> Select the Settings Code Datatypes menu option; the Programming Languages Datatypes dialog displays In the Product Name field type the name of the new language, and in the Datatype field type a datatype (one is enough to declare that the new language exists) 	Data Types <small>1987</small>
2	<p>Select the Settings Code Generation Templates menu option. The Code Templates Editor view displays.</p>	
3	<p>In the Language field, click on the drop-down arrow and select the custom language.</p>	
4	<p>From the Templates list, select one of the base templates.</p>	
5	<p>Define the template using the Code Templates Editor.</p>	
6	<p>Click on the Save button to store the template in the .EAP file.</p>	
7	<p>Repeat steps 1 to 6 for each of the relevant base templates for the custom language.</p>	

Notes

- The File template must be defined for the custom language; the *File* template can then see the *Import Section*, *Namespace* and *Class* templates

Part



7 Requirement Models

Topics

Topic	Detail	See also
Introduction	<p>This section describes the Enterprise Architect Requirements Management facilities, and discusses:</p> <ul style="list-style-type: none"> • What requirements are • How requirements are generated and organized • How Enterprise Architect supports and simplifies Requirements Management 	Requirements Management with Enterprise Architect (White Paper)
What is a Requirement?	<p>Requirements are essentially what the system, application or business process <i>is required</i> to do.</p> <p>A requirement can be:</p> <ul style="list-style-type: none"> • Broad and high level, defining - for example - that a process is necessary to update a particular database • More specialized and detailed, recording the expectation that - for example - a system call must always be acknowledged by return <p>Detailed requirements can be organized into a hierarchy culminating in a high-level requirement, so that satisfying each of the detailed requirements results in meeting the higher-level requirements and ultimately the top-level requirement.</p> <p>This hierarchical structure helps manage the complexity of large systems with thousands of requirements and many processes being developed to implement the requirements.</p>	
Gathering Requirements	<p>Gathering requirements is typically the first step in developing a solution, be it for developing a system or a process. Requirements are gathered from all parties expected to use, maintain or benefit from the solution, and are organized into groups, functional areas and hierarchies as necessary. They can be transcribed into a spreadsheet or a requirements gathering or management tool, or they can be created in an integrated modeling tool such as Enterprise Architect.</p>	
Requirements Management and Enterprise	<p>The management of requirements is one of the more problematic disciplines in software development, for reasons such as:</p>	Create Requirements ^[1475] View Requirements ^[1489]

Topic	Detail	See also
Architect	<ul style="list-style-type: none"> • Diverse group input into the requirements • Organizational boundary divisions • Tool boundary divisions • Volatility of requirements • Imprecision and ambiguity in natural languages <p>These can cause issues with:</p> <ul style="list-style-type: none"> • Traceability and • Integration with change and configuration management systems <p>Enterprise Architect can reduce or eliminate these problems in Requirements Management; it is one of the few UML tools that integrate Requirements Management with other software development disciplines in the core product, by defining requirements within the model. Within Enterprise Architect, you can:</p> <ul style="list-style-type: none"> • Create and view requirements as entities and properties directly in the model • Collate the requirements in an external CSV file and then import them into your model • Detail use cases and scenarios directly in the model • Enter standard attributes (properties) for each requirement, such as difficulty, status and type, and define your own attributes (properties) • Trace requirements to Use Cases, business rules, test cases and analysis artifacts (using, for example, the Relationship Matrix) • Trace and view the impact of changes on requirements (through, for example, the Traceability window) and review the changes themselves • Create customer-quality MS Word and HTML reports on requirements 	Import Requirements Using CSV ^[1481] Model Requirements ^[1484] Requirement Properties ^[1475] Extend Requirement Properties ^[1478] Connect Requirements ^[1480] Trace Use Of Requirements ^[1490] Relationship Matrix ^[639] Traceability Window ^[637] Manage Requirement Changes ^[1491] Report on Requirements ^[1492]

Notes

- All of these features are illustrated by examples in the *EASample.eap* model, provided as part of your

Enterprise Architect installation in the Enterprise Architect Program Files directory:

..Program Files\Sparx Systems\EA

7.1 Requirements

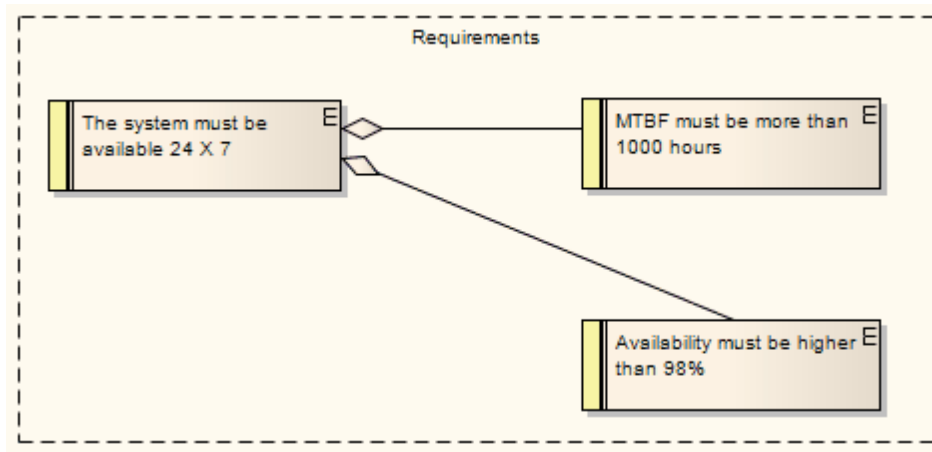
Description

As an analysis step, often it is desirable to capture simple system requirements. These are eventually realized by Use Cases.

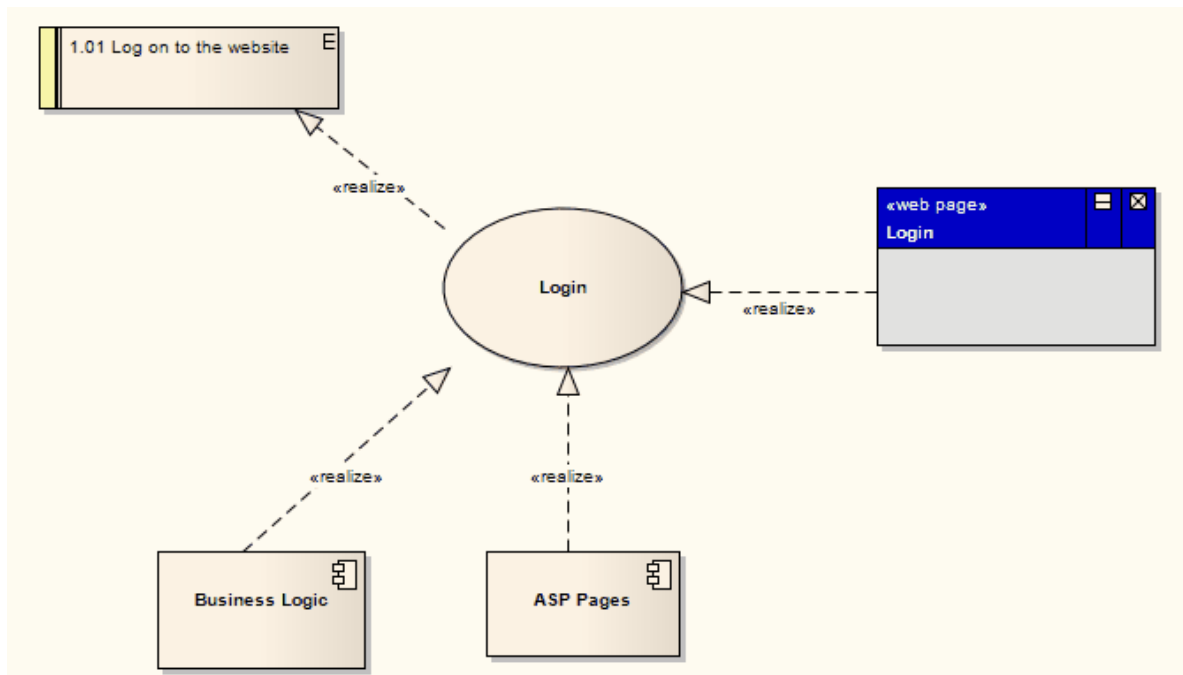
In the initial requirement gathering phase, cataloging requirements can be achieved using the Requirement extension on a Custom diagram.

Examples

Requirements can be aggregated to create a hierarchy. The diagram below illustrates how this might be done.



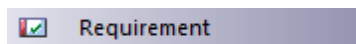
In the following diagram, a requirement that a user can log into a website is implemented by the *Login* Use Case, which in turn is implemented by the *Business Logic*, *ASP Pages* and *Login Web Page*. Using this approach, you can easily model quite detailed and complex dependencies and implementation relationships.



Notes

- External requirements can be displayed with or without an identifying **E** (for External) in the top right corner of the element; to toggle the display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the Options dialog, Objects page
- The colors on Requirement elements identify the status of the requirement; you change the status - and hence color - on the element Properties dialog, and set the color for each status on the Status Types dialog

Toolbox icon



7.2 Create Requirements

Topics

Topic	Detail	See also
Usage	<p>Within Enterprise Architect you can create external Requirement elements in a number of ways, such as:</p> <ul style="list-style-type: none"> • Dragging a <i>Requirement</i> icon from the Toolbox into a specific diagram (which also adds the Requirement to the diagram's parent package - see below) • Dragging text from a text file onto a diagram, to generate a requirement based on that text - see below • Generating an element within a specific package in the Project Browser • Importing requirements from a spreadsheet application such as Excel, via CSV • Creating Requirement elements on the Package Browser or Diagram List for the selected package or diagram • Converting an internal responsibility into an external element, in a selected target package • Importing requirements from another requirements management tool, such as <i>Telelogic DOORS</i> (in this case via the Sparx Systems <i>MDG Link For DOORS</i> integration tool) 	<p>Add Elements Directly To Packages ^[798]</p> <p>Import Requirements using CSV ^[1487]</p> <p>Package Browser ^[587]</p> <p>Diagram List ^[596]</p> <p>Converting an Internal Responsibility ^[1486]</p> <p>Sparx Systems MDG Link For DOORS</p>

How to

To create Requirement elements in a diagram

Step	Action	See also
1	Open the Custom pages on the Toolbox.	Custom Group ^[716]
2	Drag the <i>Requirement</i> element onto the current diagram.	
3	Enter the Name and other details for the requirement. Enterprise Architect creates a Requirement element in the current diagram and in the diagram's parent package.	Requirement Properties ^[1475]

Alternatively:

(The following procedure converts a text section heading into an element name and the section text into the element's Notes text.

You can use this procedure to generate elements of a *range* of types; however, it is particularly useful for generating Requirements from a requirements specification document.)

Step	Action	See also
1	Open an Enterprise Architect diagram in the Diagram View.	
2	Open the required document file containing the text you want to generate Requirement elements from (this can be opened in any common text editing tool).	
3	Highlight the required heading and associated text and drag them from the text file into the diagram. The Toolbox Shortcut menu displays.	Toolbox Shortcut Menu ^[704]
4	Navigate through the menus and select the required element type (in this case, click on Common and Requirement).	
5	Enterprise Architect creates a (Requirement) element in the diagram, and displays the Properties dialog with the section heading in the Name (or equivalent) field and the text in the Notes field; the element is also added to the diagram's parent package.	

Notes

- The Requirement element name can be simply descriptive text, with or without a manually-typed reference number; however, as requirements often have to have a unique reference for external checking, you can use the Enterprise Architect *auto-numbering* facility to automatically apply a numbering system with or without prefixes and suffixes - set the element type to **Requirement**
- External Requirement elements can be displayed with or without an identifying **E** in the top right corner; to toggle display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the Options dialog, Objects page
- Requirement elements can be color coded to indicate their status

Learn more

- [Objects](#) ^[556]
- [Use Auto Naming and Auto Counters](#) ^[799]

- [Color Code External Requirements](#)^[1477]

7.2.1 Requirement Properties

Requirement properties differ slightly from the properties of other elements; they include information related to the Type, Status, Difficulty and Priority of the Requirement. The **Notes** field is also important, as it describes precisely what requirement the element represents. Requirement naming can also require careful consideration and could reflect either a categorical naming convention, or simply a loose English description of the Requirement.

Access Requirement element context menu | Properties

Use to

- Document requirements
- Set Requirement features such as Type, Status and Priority
- Set other element properties common to both Requirements and other model elements

Reference

Field	Usage	See also
Short Description	The name of this requirement, which could include numbering, a plain English description or some other formal specification. If you have set autonaming and autonumbering, and you have already have some text in this field, it is over-written by the auto-counter text.	
Alias	An alias to be used for this requirement. If you have set Alias autonaming and autonumbering, and you have already have some text in this field, it is over-written by the auto-counter text.	Use Auto Naming and Auto Counters ^[799]
Status	The current status of this requirement.	Color Code Requirements ^[1477]
Difficulty	An estimate of the difficulty in meeting this requirement; select from: <ul style="list-style-type: none"> • Low • Medium • High 	
Priority	The relative importance of meeting this requirement compared to other requirements; select from:	

Notes	The description of this requirement.	Note Editors ^[977] Linked Documents ^[928]
--------------	--------------------------------------	--

Notes

- In Requirement Management tools and texts, the characteristics of a requirement are commonly called attributes; however, in UML the term *attribute* refers to a different type of feature, and the requirement characteristics are defined as *properties* - in this Enterprise Architect documentation, the term *properties* is used
- In a project, it might be necessary to define more information in a requirement than is provided by the standard properties; for more information on extending the requirement properties, see the *Extend Requirement Properties* topic

Learn more

- [Extend Requirement Properties](#) ^[1478]

7.2.2 Color Code External Requirements

External requirements can be color coded to provide quick visual cues indicating the status of a requirement.

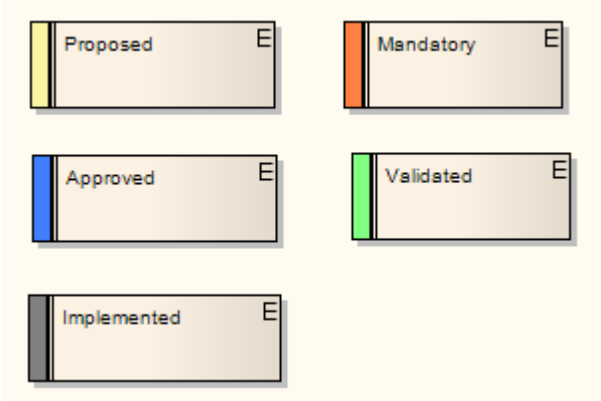
How to

To enable color coded external requirements

Step	Action
1	Select the Tools Options menu option. The Options dialog displays.
2	From the hierarchical tree select Objects , and select the Show status colors on diagrams checkbox to enable the status of external requirements to be represented by color coding.

Topics

Topic	Detail	See also
Usage	The color code requirements use the following default conventions: <ul style="list-style-type: none"> • Yellow for Proposed • Blue for Approved • Green for Validated 	Status Types Dialog ^[995]

Topic	Detail	See also
	<ul style="list-style-type: none"> • Orange for Mandatory • Black for Implemented <div style="text-align: center; margin: 10px 0;">  </div> <p>You can change these colors, and add or remove status types, using the Status Types dialog.</p>	

7.2.3 Extend Requirement Properties

Topics

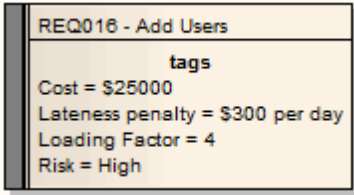
Topic	Detail	See also
Introduction	<p>A project might apply further properties to a requirement, such as cost, lateness penalty or risk to the business if not met. You can add these properties to specific Requirement elements, or configure them to be automatically available in all Requirement elements on creation, using Tagged Values. (These are sometimes referred to as User-defined attributes.)</p> <p>Extended element properties are not visible unless you open the Tagged Values window for the element. Alternatively, you can display the additional properties on the element image on its diagrams.</p>	Tagged Values ^[969] Display Tagged Values on Diagrams ^[1479]
Add Tagged Values to Existing Requirements	<p>To add a property to a single Requirement as a Tagged Value, simply click on the Requirement, display the Tagged Values window (Ctrl+Shift+6), and enter the name of the property as the tag name and the value of the property as the tag value.</p> <p>It is likely that any property you add to one Requirement would also apply to others. You might therefore use a predefined Tagged Value Type to identify your Requirement property, so that you can select it whenever required. The predefined Tagged Value Type also enables you to define specific values for the Tagged Value.</p> <p>If the appropriate predefined Tagged Value Type does not exist, a Technology Developer can create it to add to the structured tags,</p>	Assign a Tagged Value to an Item ^[972] Create Structured Tagged Values ^[1389] Create Reference Date Tagged Values ^[1392]

Topic	Detail	See also
	<p>reference tags, or customized tags collections.</p>	<p>Create a Custom Tagged Value Type ^[1393]</p>
<p>Configure Requirements to be Created With Extended Properties</p>	<p>If it is necessary to create all Requirements with the same extended set of properties, you can create a Requirement Template diagram and either create a special Requirement that defines those properties (as Tagged Values), or drag an existing Requirement with those properties onto the diagram. You then set the Requirement Template diagram as the template for all new Requirement elements, so that those new Requirements automatically have all of the properties you want.</p> <p>However, this then excludes other Requirement element formats, including the standard Requirement format. If you want to use another Requirement format, you have to replace or cancel the current Template. Alternatively, you can create a Profile.</p> <p>A Profile also defines exactly what a new Requirement element should contain, and how it should display in diagrams. However, a Profile is a collection of alternative element definitions, so it does not override the default Requirement format, nor does it prevent you from defining several different types of Requirement element. You can therefore have separate and parallel definitions of elements for business requirements, system requirements, project requirements, or any other category of requirement you decide to work with.</p> <p>For information on importing and using existing Profile files, see the <i>UML Profiles</i> topic. For information on creating new Profiles, see the <i>Developing Profiles</i> topic.</p>	<p>Set Element Template Package ^[820]</p> <p>UML Profiles ^[1277]</p> <p>Developing Profiles ^[1293]</p>

7.2.4 Display Tagged Values On Diagrams

Topics

Topic	Detail	See also
<p>Usage</p>	<p>If you have extended the properties of a Requirement, you might want to make those properties visible in the Requirement elements in your diagrams, by switching on display of the element tags compartment.</p> <p>You can do this in one of two ways:</p> <ul style="list-style-type: none"> To display the <i>tags</i> compartment on all elements on a diagram, double-click on the diagram background and select the Elements tab of the Diagram Properties dialog; select the Tags checkbox and click on the OK button To display the <i>tags</i> compartment on a specific element on a diagram, right-click on the element and select the Feature Visibility context menu option; select the Tags checkbox in the 	

Topic	Detail	See also
	<p>Show Element Compartments panel of the Feature Visibility dialog, and click on the OK button</p> <p>The Tagged Values are then displayed in the Requirement element on the diagram.</p> 	

Learn more

- [Elements](#)^[729]
- [Feature Visibility](#)^[745]

7.2.5 Connect Requirements

Topics

Topic	Detail	See also
Abstract	<p>A Requirement element can be connected to other Requirements, most commonly using Aggregate relationships to form a hierarchy of requirements.</p> <p>Requirements are also connected to other types of element, most commonly Use Cases, by Realize or Implements relationships.</p> <p>These relationships are very important, both in identifying how the Requirements are organized and used in the model, and in tracing the development from the Requirements throughout the model. Both of these tasks are very simple in Enterprise Architect, because once a connector on a Requirement exists, Enterprise Architect automatically lists the Requirement in the:</p> <ul style="list-style-type: none"> • Requirements Traceability window (an important tool in examining the role of Requirements in the model) • Requirements tab of the target element Properties dialog • Links tab of the Requirement element Properties dialog • Scenarios & Requirements window • Relationships Window • Dependency and Implementation reports • Standard RTF output 	<p>Aggregate relationships^[1208]</p> <p>Realization^[1254]</p> <p>Trace Requirements^[1490]</p> <p>Traceability^[637]</p> <p>Requirements^[848]</p> <p>Requirement Properties^[1475]</p> <p>Scenarios & Requirements^[879]</p> <p>Relationships Window^[648]</p> <p>Dependency^[2339]</p> <p>Implementation reports^[2341]</p>

Topic	Detail	See also
	The connector itself is also listed in the Links tab of the target element Properties dialog, and in the Relationship Matrix. There are, therefore, many ways to locate, view and track Requirement relationships.	Standard RTF output ^[2251] , Links ^[851] , Relationship Matrix ^[639]
Connect On Diagram	Relationships can be created on a diagram by clicking on the appropriate connector icon from the Requirement and Common pages of the Toolbox, clicking on the source (originating) element, and dragging to the target element. If you are connecting elements in different packages, you can drag elements from the Project Browser onto a common diagram and set up the relationships there.	Requirement Group ^[717] , Common Group ^[706]
Quick Generation Of Realize Connector	You can quickly generate a Realize connector by dragging an existing Requirement element from the Project Browser into a diagram, over the element that implements the Requirement (usually a Use Case). Enterprise Architect interprets this as a request to create the Realize connector and does so automatically. The Requirement element is not added to the diagram. However, if you subsequently drag the Requirement onto the diagram the connector is already in place.	
Connect Off Diagram	You can also connect a Requirement element to other elements without necessarily having the elements on the same diagram, or having a diagram open. Use the Relationship Matrix to create relationships for requirements; this is a convenient way of quickly building up complex relationships and hierarchies.	Creating Relationships ^[644]

7.2.6 Import Requirements and Hierarchies in CSV

Topics

Topic	Detail	See also
Usage	You can import Requirements from a spreadsheet application in CSV format. Before doing this you must create a CSV import file specification that: <ul style="list-style-type: none"> • In the Default Types field has the value requirement,package to import requirements and a package structure to contain them • Has the Preserve Hierarchy checkbox selected • Identifies the data fields on the spreadsheet that are to be translated into Enterprise Architect, in the order in which they 	

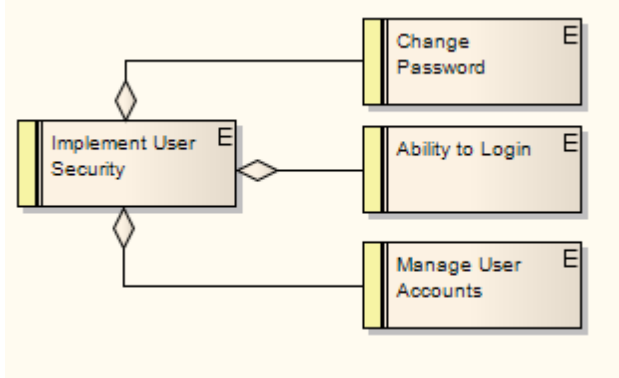
Topic	Detail	See also
	<p>are plotted across the spreadsheet</p> <ul style="list-style-type: none">• Is to operate on a spreadsheet containing the CSV_KEY and CSV_PARENT_KEY fields (which, if not generated by a CSV export from Enterprise Architect, you have added and populated yourself) <p>This enables you to import the individual and grouped requirements from the spreadsheet into Enterprise Architect, and to reconstruct the hierarchies of Requirements in the target package in the Project Browser.</p>	

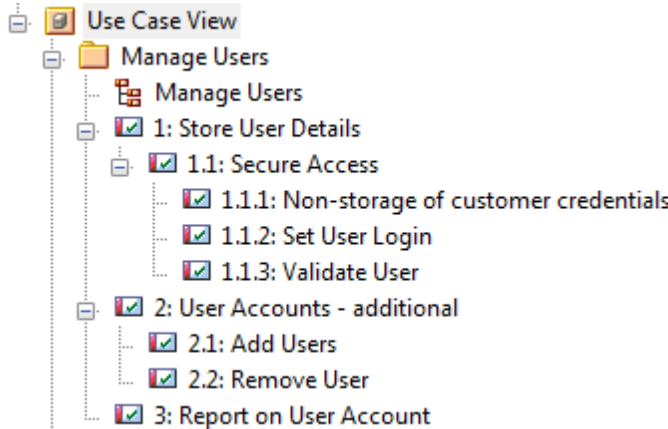
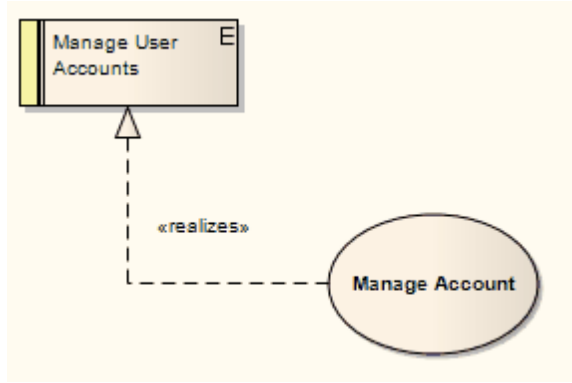
Learn more

- [CSV Import](#) ^[440]
- [CSV Specifications](#) ^[435]
- [Using Preserve Hierarchy](#) ^[438]

7.3 Model Requirements

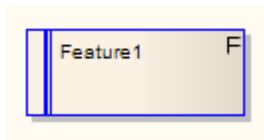
Topics

Topic	Detail	See also
Represent Requirements	<p>In Enterprise Architect, a requirement can be modeled as an:</p> <ul style="list-style-type: none"> • External Requirement - an expectation of the system or process, what the system or process must provide, modeled as an element; for example, a business requirement or a stakeholder request - Requirements at this level have their own properties and are reported on separately in RTF reports • Internal requirement – a responsibility of an existing element, what the element must do or accomplish, defined as a property of the element <p>Requirements Management in Enterprise Architect is primarily concerned with external Requirement elements and the elements that implement or realize them.</p>	Requirements <small>[147]</small> Internal requirements <small>[148]</small>
Requirements in the Model	<p>Requirement elements can be grouped and organized within Requirements diagrams.</p> <p>The Requirement elements are connected to each other by <i>Aggregate</i> relationships to form a hierarchy:</p>  <p>It is quite usual to develop a package of many hundreds of Requirement elements, arranged individually and in hierarchies of varying complexity. In the Project Browser you can use the Turn On Level Numbering option to highlight the order and arrangement of the Requirements quickly and easily.</p> <p>The following illustration shows a number of Requirements in a package, where Level Numbering makes the order and arrangement clear:</p>	Requirements Diagram <small>[149]</small> Turn On Level Numbering <small>[57]</small> RTF Style Template Editor <small>[227]</small>

Topic	Detail	See also
	 <p>If elements are added, moved or deleted from the package, the numbering automatically adjusts.</p> <p>This numbering can also be applied in the RTF report generator using the LevelNumber field in the <i>Element</i> section – {<i>Element.LevelNumber</i>}.</p>	
<p>Use Cases</p>	<p>Requirements are implemented (realized) by model elements such as Use Cases, Classes, Interfaces and Components. There are many ways to trace either the Requirement for the feature or service modeled by the elements, or the elements that develop the requirement, most visibly in Traceability diagrams that depict the Requirements and the model elements connected by Realize relationships. The Realize connector enables members of the project team to keep design objectives and development in tandem, and the development path and purpose clear.</p>  <p>The more usual realization relationship is between a Requirement and a Use Case. A Requirement can be realized by one or more Use Cases, and a Use Case can realize one or more Requirements.</p> <p>Whilst a Requirement defines a condition that must be met, the Use Case is the key to defining and visualizing how that condition is met. A Use Case diagram depicts the logical grouping of actions, processes</p>	<p>Trace [635]</p> <p>Traceability [649]</p> <p>Connector Requirement [1480]</p> <p>Use Case [1174]</p> <p>Use Case Diagram [1030]</p> <p>Actor [1107]</p> <p>Composite Element [825]</p> <p>Sequence Diagram [1074]</p> <p>Communication Diagram [1084]</p> <p>Activity Diagram [1028]</p> <p>State Machine Diagram [1032]</p> <p>Business Rule Flow [1530]</p>

Topic	Detail	See also
	<p>and components to achieve a required result, and through the use of Actor elements also defines the user and/or system roles participating in the process.</p> <p>Each Use Case (as a composite element) can contain a combination of child diagrams that define in greater detail how a particular activity or facility might be implemented - such diagrams include Sequence, Communication, Activity, State Machine and Business Rule Flow diagrams. The actual implementation of each Use Case is realized by Class, Component and Interface elements organized in their own diagrams. These realizations can also be captured and viewed in Traceability diagrams, depicting the full development pathway from initial requirement through to testing and production.</p>	

7.3.1 Feature



Description

A *Feature* is a small, granular function or characteristic expressed in client-valued terms as a satisfaction of a requirement; for example: 'context-sensitive Help', or 'ability to reverse-engineer VB.Net'.

Features are the primary requirements-gathering artifact of the Feature-Driven Design (FDD) methodology. They define the product feature that satisfies what a Requirement element has formalized as a contractual, testable, expected deliverable (for example: requirement - 'every element must provide context-sensitive Help'; feature - 'every element provides context-sensitive Help'). One Feature might realize one or more Requirements, and one Requirement might be realized by more than one Feature.

Features also have relationships with Use Cases. A Use Case defines the interaction a user has with the system in order to satisfy one or more Requirements. The Feature identifies the facility that provides the means for that interaction.

Feature elements are non-UML and are not related to the UML elements of the same name which are either BehavioralFeatures (operations, or methods) or StructuralFeatures (Ports, Parts and attributes).

Feature elements are available from the Requirements page of the Toolbox.

Feature elements can be displayed with or without an identifying **F** in the top right corner of the element. To toggle the display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the Options dialog, Objects page.

Toolbox icon



Learn more

- [Feature-Driven Design \(FDD\) methodology](#)
- [Requirement Element^{\[147\]}](#)
- [Use Case Element^{\[117\]}](#)

7.3.2 Internal Requirements

Topics

Topic	Detail	See also
Usage	<p>Internal requirements in Enterprise Architect are element responsibilities. They are defined on the Requirements page of the element Properties dialog.</p> <p>Internal requirements form the functional requirements of the system to be built. The meaning of the requirement can vary depending on which element is the host; for example, a business process requirement might mean something different to a Use Case requirement, which again might mean something different to a Class requirement.</p> <p>For example, an internal responsibility to enable the user to login to the system has been defined for the Login Use Case. This is a responsibility of the Use Case - an action it is responsible for carrying out - and it applies only to this Use Case.</p> <p>The significant parameters (or, in Requirement Management terms, <i>attributes</i>) are the Type, Status, Difficulty and Priority. Whilst you can provide a detailed description of the responsibility in the Notes field, there is more scope in the name (Requirement field) to define the nature of the responsibility. An additional field, Stability, indicates the probability of the requirement changing; high stability means a low probability of change.</p> <p>The example Use Case above also has connections to two external requirements, which are system functions that the Use Case implements either in full or in part. You can convert an internal responsibility into an external requirement.</p> <p>You can also create internal responsibilities for an element using the Scenarios & Requirements window. A responsibility created in the window displays in the element Properties dialog, and vice versa.</p>	<p>Requirements^[848]</p> <p>Make Internal Requirements External^[1486]</p> <p>Scenarios & Requirements^[879]</p>

7.3.2.1 Make Internal Requirement External

Elements in Enterprise Architect have internal requirements, or responsibilities (what they must do or accomplish). These often overlap or duplicate more formal requirements that the system in general must meet. You can move internal requirements to external requirements (where the requirement can perhaps be implemented by multiple elements) in one go, using the **Move External** function.

How to

To change an internal requirement for an element into an external element

Step	Action	See also
1	Double-click on the element in a diagram or in the Project Browser. The element Properties dialog displays.	
2	Click on the Requirements tab.	
3	Locate and highlight the requirement.	
4	Click on the Move External button. The Browse Project dialog displays.	
5	Select the package to place the new requirement in.	
6	Click on the OK button. Enterprise Architect creates a new Requirement element in the target package and automatically generates a Realization connector from the element to the Requirement. The requirement is now marked <i>external</i> and the dialog fields grayed out. To edit its details, double-click on the requirement. The requirement element is now located in the target package.	

Learn more

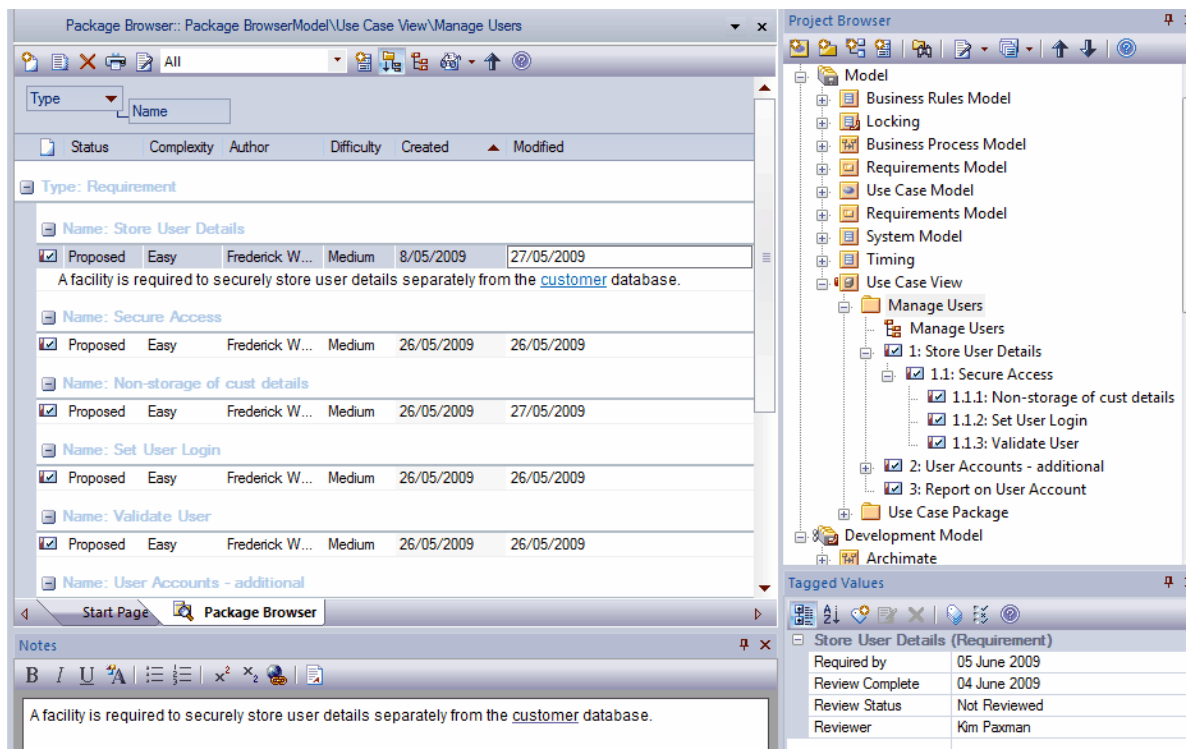
- [Requirement Properties](#) 

7.4 Manage Requirements

One of the main advantages of managing Requirements in Enterprise Architect is that you can display many different aspects of information on the Requirements, from overall organization and location through lists, current status, general properties, detailed individual notes and specific properties, to relationships and dependencies with other elements in the model.

As most of these aspects are displayed in dockable windows, you can review the Requirements from several different perspectives simultaneously in the Enterprise Architect work area, as shown below:

Example



This display shows the position of the *Store User Details* Requirement element in the model, and how it relates to other Requirements (Project Browser); the default characteristics of the Requirement (Package Browser) and the extended characteristics (Tagged Values window), and a detailed description of the Requirement (Notes window). You can configure some of these windows to display more information, and/or use other windows and facilities.

Learn more

- [View Requirements](#) ^[1489]
- [Trace Use of Requirements](#) ^[1490]
- [Manage Requirement Changes](#) ^[1491]
- [Report on Requirements](#) ^[1492]

7.4.1 View Requirements

Use the following windows and facilities to: locate and list Requirement elements in the model; add, move and delete the elements; display and edit the properties and characteristics of individual elements; and generate reports on packages or specific elements.

Topics

Topic	Detail	See also
Usage	<p>Use the following windows and facilities to: locate and list Requirement elements in the model; add, move and delete the elements; display and edit the properties and characteristics of individual elements; and generate reports on packages or specific elements.</p> <ul style="list-style-type: none"> • Project Browser - shows the content and structure of your model • Diagram List - lists the elements in a diagram, filtered and sorted according to the settings you define; shows all or selected default properties of each element • Package Browser - lists the elements in a package, filtered and sorted according to the settings you define; shows all or selected default properties of each element • (Requirements) Diagram - shows the arrangement of a group of Requirements, and can show whether the elements are in the same package or different packages • Model Search - enables you to locate Requirements in general in the model, or specific Requirement elements, according to the search criteria you use • Model Views - enables you to maintain links to commonly-used elements, and to rapidly show developments and changes in (Requirement) package contents through either reports or slide shows of selected diagrams • Properties - shows every standard property of a selected element, whether updated by the user or maintained automatically by the system • Tagged Values - shows extended properties of a selected Requirement element • Element Browser - shows every added-on property, such as attributes, operations, Tagged Values and constraints • Notes - displays the detailed description of a requirement, and any other additional information recorded on the requirement 	<p>Project Browser ^[567]</p> <p>Diagram List ^[596]</p> <p>Package Browser ^[587]</p> <p>Requirements Diagram ^[1494] ^[69]</p> <p>Model Search ^[61]</p> <p>Model Views ^[598]</p> <p>Properties ^[878]</p> <p>Tagged Values ^[969]</p> <p>Element Browser ^[876]</p> <p>Notes ^[977]</p>

7.4.2 Trace Use of Requirements

Topics

Topic	Detail	See also
Usage	<p>Having investigated the representation of requirements in your model, you might review either how they have been used to direct development through the model, or how a particular development was initiated.</p> <p>This is discussed in greater detail in the <i>Traceability</i> topics, but the windows and facilities you might use to follow development from Requirements are briefly described below.</p> <p>The significant feature in tracing Requirements and development is the connectors between the elements.</p> <ul style="list-style-type: none"> • Relationships window - quickly identifies every relationship of which a selected Requirement element is a member, and the partner element in that relationship, whether or not the relationship is visible in the current diagram. If the partner element is not in the diagram, you have the option of adding it • Traceability window - a very useful tool in showing chains of relationships that include the selected element. The window can show, for example, that a Requirement A is realized by a Use Case X, but Use Case X also realizes Requirement B, which in turn is also realized by Use Case Y. You can control the type and extent of these relationship chains, but as the system checks the connectors and partner elements of every relationship within the limits you impose, the system can take some time to produce the final results • Relationship Matrix - a significant tool in mapping the relationships between the Requirements elements in a package and other elements in either that package or a different package. Where a relationship is missing, you can add it; if an existing relationship is misplaced, you can delete it • Properties dialog, Requirements tab - for elements other than Requirements (particularly Use Cases), shows all internal responsibilities and external requirement elements attached to the element • Scenarios & Requirements window - as for the Properties dialog, shows the Requirements and responsibilities of the selected element, and the scenarios and constraints under which the Requirements are being realized 	Traceability ^[635] Connect Requirements ^[1480] Relationship Window ^[648] Traceability window ^[637] Relationship Matrix ^[639] Properties Dialog ^[843] Requirements ^[848] Scenarios & Requirements ^[879]

Topic	Detail	See also

7.4.3 Manage Requirement Changes

Because requirements are statements of what a system or process must do or provide, they have a great impact on the modeling and development of the system. A new requirement might initiate an extensive program of work, and changes to or removal of that requirement can therefore have a major effect on the model. Issues concerning requirements, and changes to Requirements, must both be carefully managed.

The first steps in managing changes to requirements would be to raise *specific Issue and Change* request items against the Requirement element. You could monitor the appearance of these items using the filtered searches of the *Model Views*. You might then **review** the Requirement properties and/or its relationship hierarchies. During model development, you might capture periodic *Baselines* and use these to review the changes and, if necessary, roll them back to a previous point. You might also use the *Auditing* facility to monitor changes as they are made, and to ensure that no unauthorized or potentially risky changes are being made in the model.

These facilities are briefly discussed below.

Topics

Topic	Detail	See also
Changes and Issues	<p>A change is, very broadly, an item defining an addition or alteration to a requirement. An issue identifies either a failure to meet a requirement, or a risk in meeting the requirement.</p> <p>Changes and issues can arise in development at a number of levels, being raised for problems that apply system-wide down to within a specific element. There are two mechanisms that can be used to identify a change or issue, and the work required to resolve it:</p> <ul style="list-style-type: none"> • Change and Issue (or Defect) elements - structured comments that identify a problem at system-level, although they can also be attached to a specific element from which a problem arises. Both types of element resemble the Requirement element, and can be linked to one or more other elements that have to be reviewed, with relationships such as Association, Dependency and Realize. The two types of element can also form hierarchies or groups, where complex problems arise • Maintenance items raised against a specific element, and recorded for that element in the Maintenance window. Maintenance items enable the distinction between Defects (a failure to meet a requirement) and Issues (a risk factor that might affect satisfying the requirement). They also include Tasks, which record work items associated with the element <p>Maintenance items are very specific, but if there is a possibility of an item having a wider impact on other elements or the system in</p>	<p>Changes ^[2246]</p> <p>Issues ^[2244]</p> <p>Maintenance items ^[2236]</p> <p>Maintenance Workspace ^[2237]</p> <p>Create an Element From a Maintenance Item ^[2241]</p>

Topic	Detail	See also
	<p>general, you can translate the item into a Change or Issue element, or any other type of element that best identifies the problem and its solution.</p>	
Model Views	<p>Model Views are very useful for trapping changes and issues in the model, especially on Requirements. You can set up searches to identify the appearance of new Change or Issue elements, or to detect changes in the properties of the Requirement elements themselves.</p>	Model Views ^[598]
Baselines	<p>A Baseline is a snapshot of a package or a model branch at a particular point in time, which you determine. You can use the Baseline as a distribution mechanism for changes to the model, but the main use is to enable you to compare the current model with a previous stage, and detect what changes have been made since the Baseline was captured.</p> <p>If you do not want a change to remain in the model, you can roll the affected elements back to the state they had in the Baseline. Therefore, if you maintain your requirements in a specific package or branch, you can capture Baselines of the package and ensure that changes conform to your change management process or, if not, can be reversed.</p>	Baselines ^[396]
Auditing	<p>The Auditing facility enables you to capture any changes made to your model within the selection criteria that you define. You can, for example, configure the Auditing facility to specifically record changes to Requirement elements.</p> <p>As auditing is continuously monitoring, you can detect changes as they are made, and verify that they are acceptable. You can also store the log of changes, and review it later on.</p> <p>Note that you cannot reverse the changes automatically, as you can with Baselines. You might therefore use Auditing to identify changes to investigate more fully and - if necessary - reverse in a Baseline comparison.</p>	Auditing ^[382] Auditing Settings ^[384]

Learn more

- [View Requirements](#) ^[1489]

7.4.4 Report on Requirements

Topics

Topic	Detail	See also
Usage	<p>Enterprise Architect provides two report generation facilities that enable you to output RTF reports and HTML reports on your model structure and components.</p>	RTF Document ^[2251] RTF Dialog

Topic	Detail	See also
	<p>The RTF reporting facility is especially comprehensive, and contains a number of features that provide particular support to Requirements Management:</p> <ul style="list-style-type: none"> • A requirements report template that extracts details of external requirements in the model; you can copy and tailor this template for your particular requirements • Options in the Diagram List, Package Browser and Model Search to generate RTF reports on selected (Requirement) items from the collected information • The Implementation Report, which lists for a selected package the elements that require implementers, together with any source elements in Realize (Implements) relationships with those elements • The Dependency Report, which lists for a selected package any elements that are dependent on another element for their specification; for example, a Use Case derives its specification from the Requirement that it realizes 	<p>Options^[2257]</p> <p>Diagram List^[596]</p> <p>Package Browser^[587]</p> <p>Model Search^[611]</p> <p>Implementation Report^[2341]</p> <p>Realization^[1254]</p> <p>Dependency Report^[2339]</p>

7.5 Requirements Diagram

A Requirements diagram is a custom diagram used to describe a system's requirements or features as a visual model.

Requirements are defined using Requirement elements (Custom elements of type Requirement). To view the detailed description of a Requirement, double-click on the element to display its properties.

Requirements can have relationships with other elements such as other Requirements and Use Cases. To view the traceability of a requirement, use the Traceability window.









Use to

- Describe a system's requirements or features as a visual model
- Link Requirement elements to Use Cases and Components in the system to illustrate how a particular system requirement is met
- Enable traceability between specifications and design requirements, and the model elements that realize them

Example Diagram [Example Requirements Diagram](#)^[1494]

Tools

Select Requirements diagram elements and connectors from the Requirements pages of the Toolbox.

Requirements Diagram Elements	Requirements Diagram Connectors
 Package	 Aggregate
 Requirement	 Inheritance
 Feature	 Associate
 Object	 Implements

Learn more

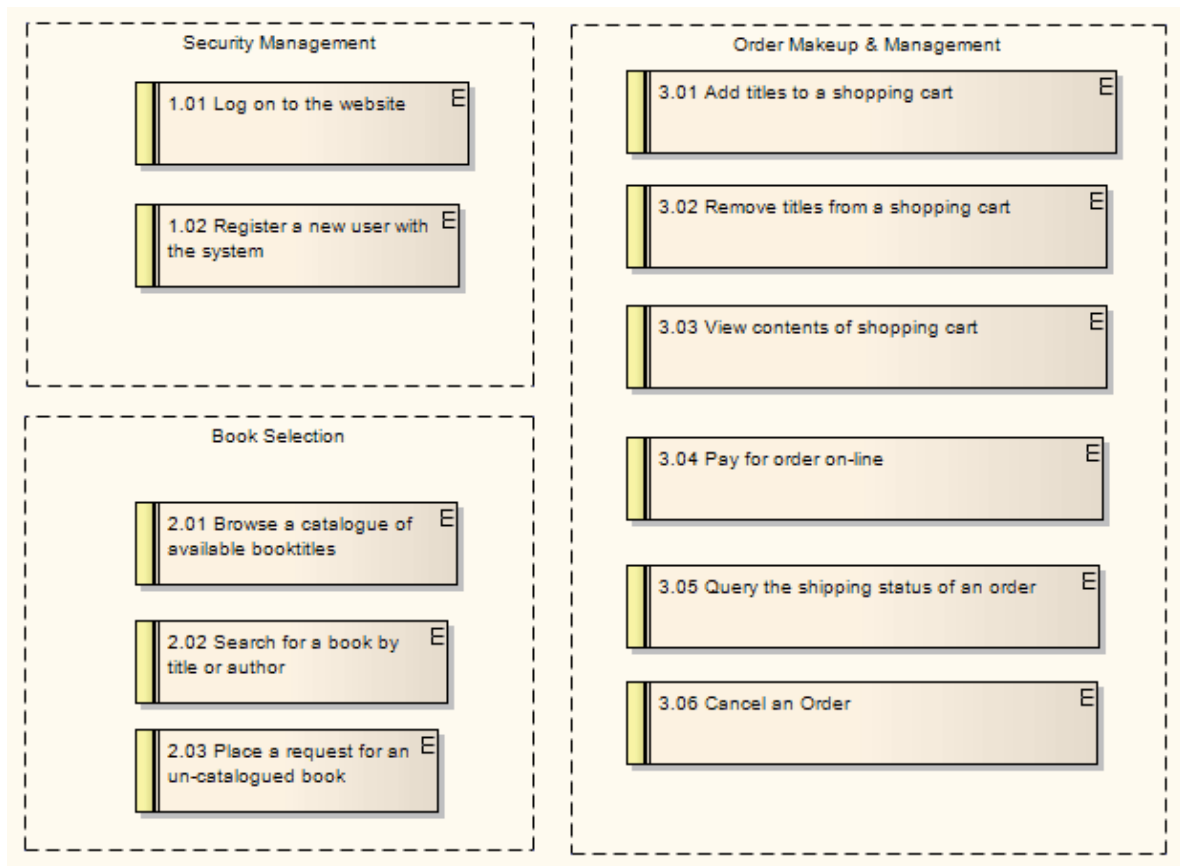
- [Trace - Tracking Dependencies](#)^[635]
- [The Traceability Window](#)^[637]

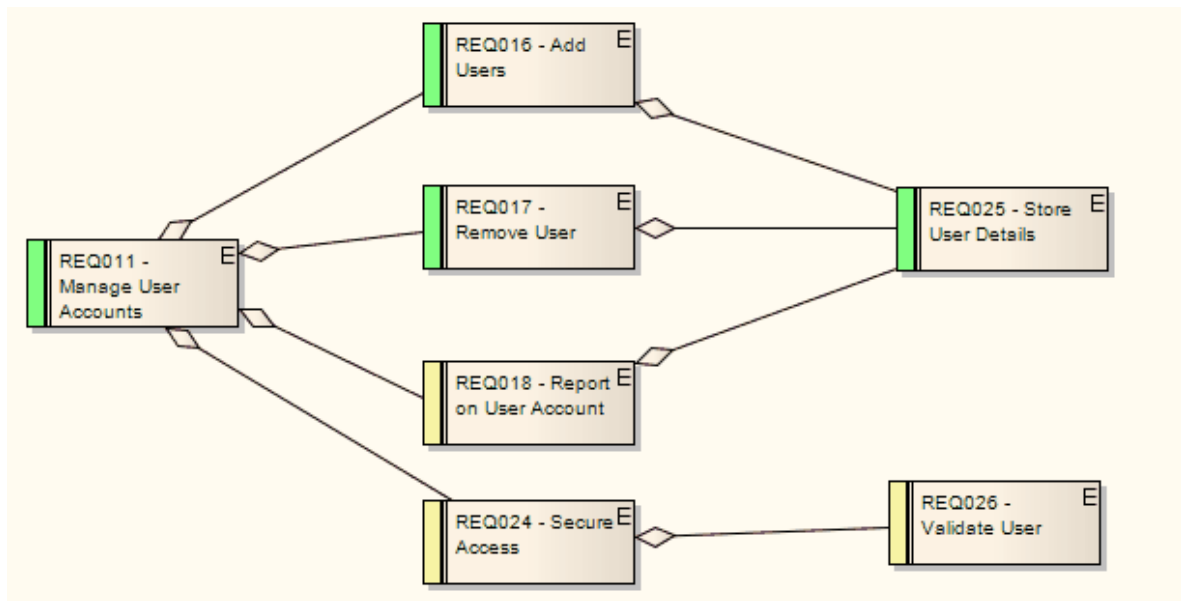
7.5.1 Example Requirements Diagram

The example below illustrates a Requirements diagram.

Requirement elements can be linked back to Use Cases and Components in the system to illustrate how a particular system requirement is met.

Change and Defect (Issue) elements look the same as Requirement elements and can be coded and managed in the same way.

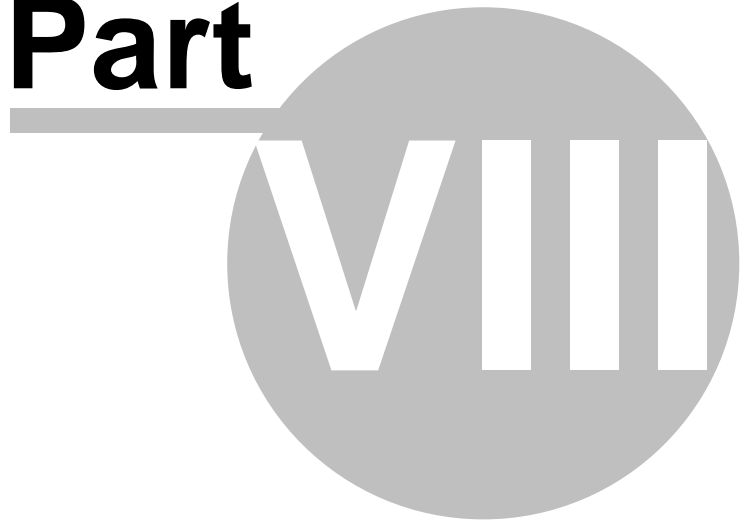
Example 1:**Example 2:**



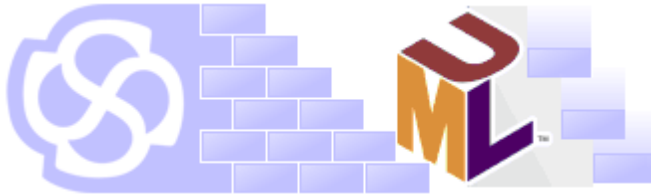
Learn more

- [Requirements Diagram](#) ^[1494]
- [Requirements](#) ^[1471]
- [Use Case](#) ^[1174]
- [Component](#) ^[1187]
- [Changes](#) ^[2246]
- [Defects \(Issues\)](#) ^[2244]

Part



8 Domain Based Models



Enterprise Architect provides specific modeling tools for a range of specialized model types, as outlined below.

Enterprise Architect also provides an additional set of extended diagrams and a set of inbuilt and extension stereotype elements for more general use, and supports you in developing your own modeling languages.

Topic	Detail	See also
ODM	Enterprise Architect enables you to develop large-scale ontologies within the fully-integrated modeling environment, for your project domain. ODM helps you to develop a formalized representation of business semantics and taxonomies, and a knowledge representation based on those formalizations.	MDG Technology for ODM <small>[1643]</small>
Requirements	Enterprise Architect is one of the few UML tools that integrate Requirements Management with other software development disciplines in the core product, by defining requirements within the model.	Requirements modeling <small>[1468]</small>
Business Modeling	Modeling the business process is an essential part of any software development process, enabling the analyst to capture the broad outline and procedures that govern what it is a business does.	Business modeling <small>[1515]</small>
Business Rules	Business Rule modeling captures the rules that govern a business, and their relationships with the entities and specific tasks within the organization or system.	Business Rule modeling <small>[1522]</small>
BPMN	The Business Process Modeling Notation is specifically targeted at the business modeling community and has a direct mapping to UML through BPMN Profiles; these profiles enable you to develop BPMN diagrams quickly and simply.	BPMN modeling <small>[1545]</small>
BPEL	Business Process Execution Language is an executable language for specifying interactions with Web Services. Enterprise Architect uses the BPMN profile as a graphical front-end to capture BPEL Process	BPEL modeling <small>[1570]</small>

	descriptions.	
SysML	SysML is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that might include hardware, software, information, personnel, procedures and facilities.	Systems engineering (SysML) ^[1964]
Data Modeling	Enterprise Architect provides easy-to-use tools for building and maintaining all of the fundamental data models - Conceptual, Logical and Physical; because Enterprise Architect lets you visualize each type of data model in the same repository, you can easily manage dependencies between each level of abstraction.	Data Models ^[1625]
XSD	Enterprise Architect supports rapid modeling, forward engineering and reverse engineering of W3C XML schemas (XSD), critical for the development of a complete Service Oriented Architecture (SOA).	XML Schema modeling (XSD) ^[2027]
WSDL	Enterprise Architect enables rapid modeling, forward engineering and reverse engineering of W3C XML Web Service Definition Language (WSDL), critical for the development of a complete Service Oriented Architecture (SOA).	Web Service modeling (WSDL) ^[2067]
SPEM	The Software and Systems Process Engineering Meta-model (SPEM) is a conceptual framework for modeling, documenting, presenting, managing, interchanging, and enacting development methods and processes. SPEM 2.0 focuses on providing the additional information structures that you require for processes modeled with UML 2 Activities or BPMN/BPDM.	Software Process modeling (SPEM) ^[523]
ArchiMate	ArchiMate is an open-standard enterprise architecture language based on the IEEE 1471 standard, providing a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems and technical infrastructure. It enables Enterprise Architects to clearly describe, analyse and visualize the relationships among business domains.	ArchiMate ^[1621]
Data Flow Diagrams	A data flow diagram (DFD) is a graphical representation of the flow of data through an information system, and can also be used to visualize data processing (structured design).	Data Flow Diagrams ^[1506]

	Developing a DFD helps in identifying the transaction data in the data model.	
Entity Relationship Diagrams	Entity-relationship modeling is an abstract and conceptual database modeling method, used to produce a schema or semantic data model of, for example, a relational database and its requirements, visualized in Entity-Relationship Diagrams (ERDs). ERDs in Enterprise Architect assist you in building conceptual data models through to generating Data Definition Language (DDL) for the target DBMS.	Entity Relationship Diagrams ^[1627]
Eriksson-Penker Extensions	Eriksson-Penker extensions provide a framework for UML business processing model extensions, to which an Enterprise Architect can add stereotypes and properties appropriate to their business.. In Enterprise Architect, the Eriksson-Penker profile provides, through a set of stereotypes, a unique and powerful means of visualizing and communicating business processes and the necessary flow of information within an organization.	Eriksson-Penker Extensions ^[1623]
Group of Four Patterns	Gang of Four (GoF) Patterns are 23 classic software design patterns providing recurring solutions to common problems in software design. Enterprise Architect provides each pattern through an icon in the Diagram Toolbox.	GoF Patterns ^[1923]
ICONIX	The ICONIX Process is a streamlined approach to Use Case driven UML modeling that uses a core subset of UML diagrams and techniques to provide thorough coverage of object-oriented analysis and design. Its main activity is robustness analysis, a method for bridging the gap between analysis and design.	ICONIX ^[1925]
Mind Mapping	A Mind Map is an image-centered diagram used to represent semantic or other connections between words, ideas, tasks or other items arranged radially around a central key word or idea. A Mind Map is used to generate, visualize, structure and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing.	Mind Mapping ^[1503]
SoaML	Service Oriented Architecture (SOA) is an architectural paradigm for defining how people, organizations and systems provide and use services to achieve results.	SoaML ^[2088]

SOMF	The service-oriented modeling framework (SOMF) is a service-oriented development life cycle methodology, offering a number of modeling practices and disciplines that contribute to a successful service-oriented life cycle management and modeling.	SOMF ^[2092]
Extended Diagrams	Enterprise Architect provides an additional set of diagram types that extend the core UML diagrams for domain-specific models. Also, the specialized modeling tools listed in the first part of this table each have their own specialized diagrams.	Analysis Diagram ^[1509] Custom Diagram ^[1505] Requirements Diagram ^[1494] Maintenance Diagram ^[2234] User Interface Diagram ^[1655] Database Diagram ^[1631] Business Modeling and Business Interaction Diagrams ^[1514]
Inbuilt and Extension Stereotypes	Behavioral and Structural elements can be extended through the use of stereotypes; Enterprise Architect provides a number of inbuilt extensions.	Inbuilt and Extension Stereotypes ^[1663]
Build Your Own Modeling Language	Enterprise Architect enables you to extend the scope both of your modeling and of the UML components you use, through the use of stereotypes, profiles and patterns to develop your own modeling applications.	Build Your Own Modeling Language ^[1292]

8.1 Domain Based Diagrams

In addition to diagrams defined by the UML, Enterprise Architect provides some extended diagram platforms to model business processes or develop custom diagrams.

Diagram Type	Detail	See also
Analysis Diagram	An Analysis diagram is a simplified Activity diagram, which is used to capture high level business processes and early models of system behavior and elements.	Analysis Diagram ^[1505]
Custom Diagram	A Custom diagram is an extended Class diagram that is used to capture requirements, user interfaces or custom-design models.	Custom Diagram ^[1505]
Requirements Diagram	A Requirements diagram is a custom diagram used to describe a system's requirements or features as a visual model.	Requirements Diagram ^[1494]
Maintenance Diagram	A Maintenance diagram is a custom diagram used to describe change requests and issue items within a system model.	Maintenance Diagram ^[2234]
User Interface Diagram	User Interface diagrams are custom diagrams used to visually mock-up a system's user interface using forms, controls and labels.	User Interface Diagram ^[1655]
Data Modeling Diagram	A Data Modeling diagram is a Class diagram used for representing database schemas.	Data Modeling Diagram ^[1631]
Documentation	Virtual documents enable you to structure and filter your RTF and HTML reports by selecting, grouping and ordering individual packages independent of the organization of the Project Browser.	Documentation ^[2327]
Business Modeling and Business Interaction	<p>Business Modeling diagrams and Business Interaction diagrams enable you to model both the structure and behavior of a business system.</p> <p>Business Modeling diagrams are based on a Class (UML Structural) diagram, whilst Business Interaction diagrams are based on a Sequence (UML Behavioral) diagram.</p>	Business Modeling and Business Interaction ^[1514]

Learning Center topics

- (Alt + F1) | **Modeling Languages**

8.2 Analysis Models

See:

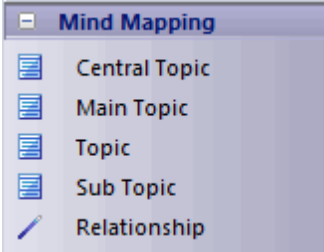
- [Mind Mapping](#) ^[1503]
- [Custom Diagram](#) ^[1505]
- [Data Flow Diagrams](#) ^[1506]
- [Analysis Diagram](#) ^[1509]
- [Analysis Stereotypes](#) ^[1509]
- [Boundary](#) ^[1663]
- [Control](#) ^[1665]
- [Entity](#) ^[1666]
- [Event](#) ^[1667]
- [Process](#) ^[1674]

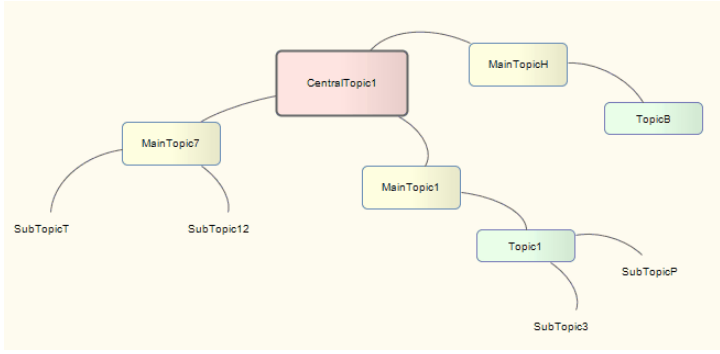
8.2.1 Mind Mapping

[Access](#) [View](#) | [Diagram Toolbox: More Tools](#) | [Mind Mapping](#)

Topics

Topic	Detail	See also
Introduction	<p>The following text is derived from the Mind Map entry in the online Wikipedia.</p> <p><i>A Mind Map is a diagram used to represent words, ideas, tasks or other items linked to and arranged radially around a central key word or idea. It is used to generate, visualize, structure and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing.</i></p> <p><i>A Mind Map is an image-centered diagram that represents semantic or other connections between portions of information. By presenting these connections in a radial, non-linear graphical manner, it encourages a brainstorming approach to any given organizational task, eliminating the hurdle of initially establishing an intrinsically appropriate or relevant conceptual framework to work within.</i></p> <p><i>The elements are arranged intuitively according to the importance of the concepts and are organized into groupings, branches, or areas. The uniform graphic formulation of the semantic structure of information on the method of gathering knowledge, may aid recall of existing memories.</i></p> <p>The use of the term Mind Maps is trademarked in the UK and the USA by The Buzan Organization, Ltd.</p>	Mind Map (Online Resource)
Mind Mapping in Enterprise	Enterprise Architect enables you to develop Mind Maps quickly and simply, through use of an MDG Technology integrated with	New Diagram ^[724]

Topic	Detail	See also
Architect	<p>the Enterprise Architect installer. The Mind Mapping facilities are provided in the form of:</p> <ul style="list-style-type: none"> • A Mind Mapping diagram type, accessed through the New Diagram dialog • A Mind Mapping page in the Toolbox • Mind Mapping element and relationship entries in the Toolbox Shortcut Menu and Quick Linker 	<p>Toolbox Shortcut^[70]</p> <p>Quick Linker^[79]</p>
Mind Mapping Toolbox Page	<p>The following icons are available:</p>  <ul style="list-style-type: none"> • <i>Central Topic</i> is the main theme of the Mind Map; you would normally have one or two of these on the diagram, but can add as many as are necessary • <i>Main Topic</i> represents the immediate concepts generated by the Central Topic • <i>Topic</i> represents the larger divisions of a Main Topic • <i>Sub Topic</i> represents the finer divisions of a Topic or Main Topic; you could also have Subtopics of Subtopics to represent increasingly finer distinctions • <i>Relationship</i> represents the connection between any two elements; you can have several Relationships per element. Each relationship has three anchor points, so you can curve the lines to develop the flow of concepts more easily <p>When dragged onto a Mind Mapping diagram, the elements and relationship have the following appearances:</p>	

Topic	Detail	See also
	 <p>As the elements can represent any concept, object or relationship, you can use the full range of element properties and features to expand on what the element represents, including adding Note elements. However, to preserve the simplicity and readability of the diagram itself, you cannot display the element compartments on the diagram.</p>	
Disable Mind Mapping	If you prefer not to use Mind Mapping in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).	MDG Technologies [1285]

Learning Center topics

- (Alt + F1) | **Modeling Languages | Standard Profiles | Mind Mapping**









8.2.2 Custom Diagram






A Custom diagram is an extended Class diagram that is used to capture requirements, user interfaces or custom-design models.

Custom models provide a few extensions to the UML model and enable some exploratory and non-rigorous experimentation with model elements and diagrams.

Tools

Select Custom diagram elements and connectors from the Custom pages of the Toolbox.

Custom Diagram Elements	Custom Diagram Connectors
 Package	 Associate
 Requirement	 Aggregate
 Issue	 Generalize
 Change	 Realize

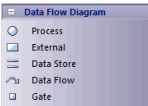
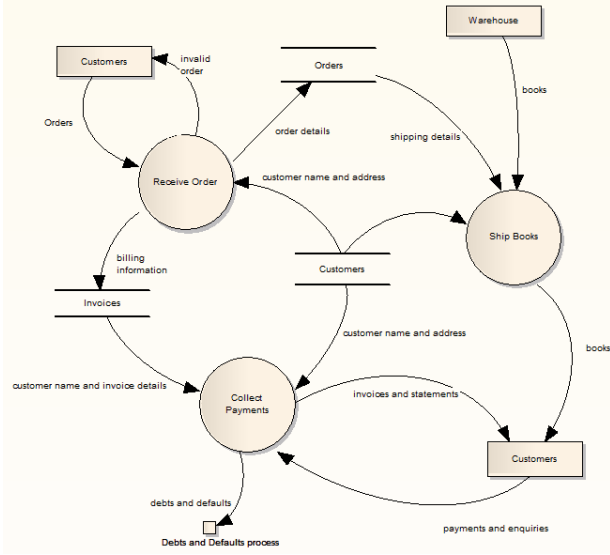
Custom Diagram Elements	Custom Diagram Connectors
 Screen	 Nesting
 UI Control	
 Test Case	
 Entity	

8.2.3 Data Flow Diagrams

Access [View | Diagram Toolbox: More Tools | Data Flow Diagrams](#)

Topics

Images	Topic	Detail	See also
	Introduction	<p>The following text is derived from the Data Flow Diagram entry in the online Wikipedia.</p> <p><i>A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then "exploded" to show more detail of the system being modeled.</i></p> <p><i>Data flow diagrams were invented by Larry Constantine ... based on Martin and Estrin's "data flow graph" model of computation. (They) are one of the three essential perspectives of Structured Systems Analysis and Design Method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a dataflow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. The old system's dataflow diagrams can be drawn up and compared with the new system's dataflow diagrams to draw comparisons to implement a more efficient system.</i></p> <p><i>Developing a DFD helps in identifying the transaction data in the data model.</i></p>	Data Flow Diagram (Online Resource)
	Data Flow Diagrams in Enterprise Architect	Enterprise Architect enables you to develop Data Flow diagrams quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer. The Data Flow diagram facilities are provided in the form of:	New Diagram Toolbox Shortcut Quick Linker

Images	Topic	Detail	See also
		<ul style="list-style-type: none"> • A Data Flow diagram type, accessed through the New Diagram dialog • A Data Flow Diagram page in the Toolbox • Data Flow element and relationship entries in the Toolbox Shortcut Menu and Quick Linker. 	
	<p>Data Flow Diagram Toolbox Page</p>	<p>The following icons are available:</p> <ul style="list-style-type: none"> • <i>Process</i> is a process or activity in which data is used or generated • <i>External</i> represents an external source, user or depository of the data • <i>Data Store</i> represents an internal physical or electronic repository of data, into and out of which data is stored and retrieved • <i>Data Flow</i> (connector) represents how data flows through the system, in physical or electronic form • <i>Gate</i> represents the termination point of incoming and outgoing messages on a lower level diagram (that is, messages to and from processes depicted elsewhere) <p>When dragged onto a Data Flow diagram, the elements and relationship have the following appearances:</p> 	

Images	Topic	Detail	See also
		To preserve the simplicity and readability of the diagram, you cannot display the element compartments on the diagram.	
	Context Diagram	A <i>Context</i> diagram is a top-level Data Flow diagram that has just one Process element representing the system being modeled, showing its relationship to external systems.	
	Disable Data Flow Diagrams	If you prefer not to use Data Flow Diagramming in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).	MDG Technologies ^[1285]

Learning Center topics

- (Alt + F1) | [Modeling Languages](#) | [Standard Profiles](#) | [Data Flow Diagrams](#)

8.2.4 Analysis Stereotypes

Description

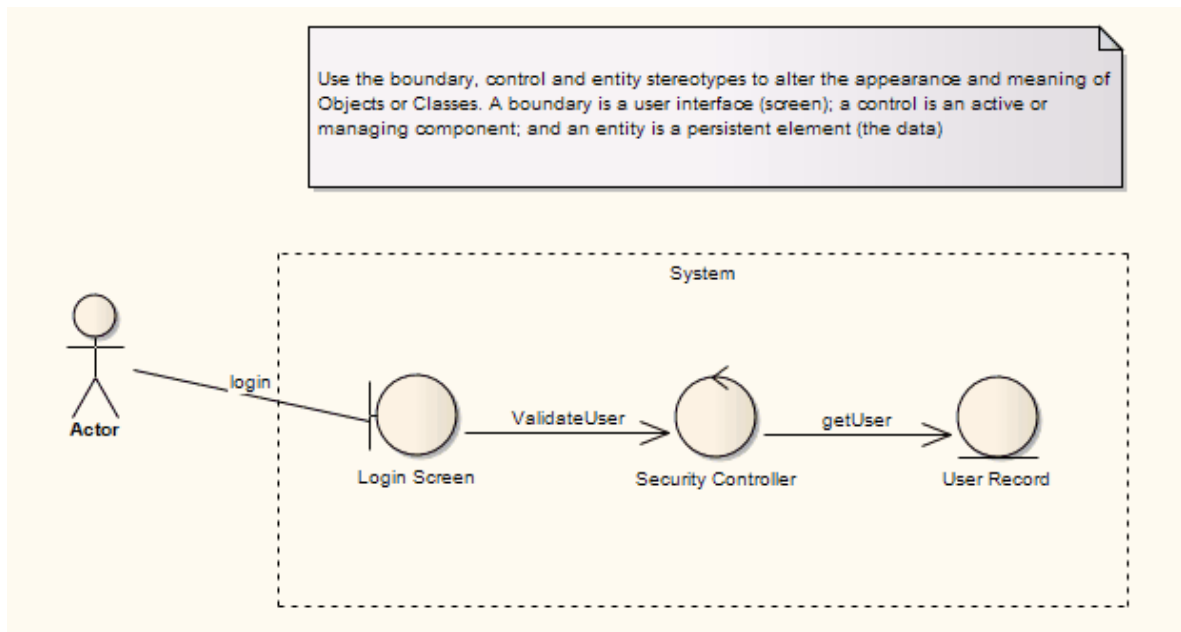
Enterprise Architect has some built in stereotypes that you can assign to an element during analysis. The effect of these stereotypes is to display a different icon from the normal element icon, providing a visual key to the element purpose. The *Robustness* diagram below illustrates the main types of inbuilt icons for elements.

The stereotypes used are:

- Boundary - for a system boundary (for example, a Login screen)
- Control - to specify an element is a controller of some process (as in the Model-View-Controller pattern)
- Entity - the element is a persistent or data element

Also see the Business Modeling elements, used in Business Modeling and Business Interaction diagrams.

Example



Learn more

- [Boundary](#)^[1663]
- [Control](#)^[1665]
- [Entity](#)^[1666]
- [Business Modeling](#)^[1514]

Learning Center topics

- **(Alt + F1) | Modeling Languages | Business | Business Modeling and Interactions.**

8.2.5 Analysis Diagram



An Analysis diagram is a simplified Activity diagram, which is used to capture high level business processes and early models of system behavior and elements. It is less formal than some other diagrams, but provides a good means of capturing the essential business characteristics and requirements.


















Robustness diagrams, used extensively in ICONIX, can be created as Analysis diagrams.

Example Diagram [Example Analysis Diagram](#)^[1511]

Tools

Select Analysis diagram elements and connectors from the Analysis pages of the Toolbox.

Analysis Diagram Elements	Analysis Diagram Connectors
 Actor	 Information Flow

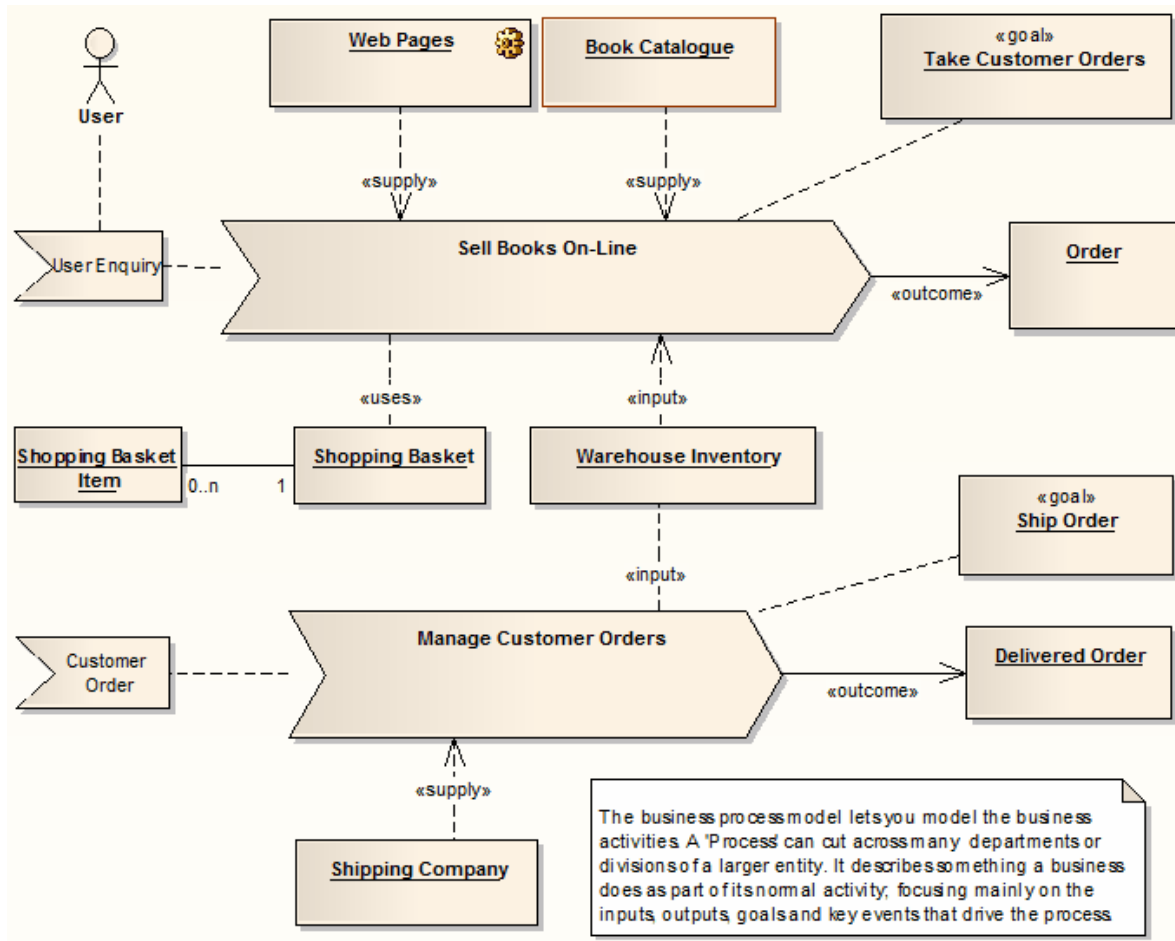
Analysis Diagram Elements	Analysis Diagram Connectors
 Object	 Object Flow
 Process	 Associate
 Collaboration	 Realize
 Collaboration Use	 Representation
 Send	
 Receive	
 Information	
 Information Item	
 Decision	
 Merge	
 Boundary	
 Control	
 Entity	

Learn more

- [Activity Diagram](#) ^[1028]
- [Business Process Modeling](#) ^[1515]
- [ICONIX](#) ^[1925]

8.2.5.1 Example Analysis Diagram

Enterprise Architect supports some of the Eriksson-Penker Business Extensions that facilitate business process modeling. The complete Eriksson-Penker Business Extensions UML Profile can also be loaded into Enterprise Architect and used to create detailed process models.



Learn more

- [Eriksson-Penker Business Extensions](#) ¹⁶²³

8.3 Business Models

A *Business Process model* describes both the behavior and the information flows within an organization or system. As a model of business activity, it captures the significant events, inputs, resources, processing and outputs associated with relevant business processes.

Enterprise Architect provides specific modeling tools for a range of analysis and Business Modeling types.

Topic	Detail	See also
Requirements	Enterprise Architect is one of the few UML tools that integrate Requirements Management with other software development disciplines in the core product, by defining requirements within the model.	Requirements modeling ^[1468]
Business Modeling	Modeling the business process is an essential part of any software development process, enabling the analyst to capture the broad outline and procedures that govern what it is a business does.	Business modeling ^[1515]
Business Rules	Business Rule modeling captures the rules that govern a business, and their relationships with the entities and specific tasks within the organization or system.	Business Rule modeling ^[1522]
BPMN	The Business Process Modeling Notation is specifically targeted at the business modeling community and has a direct mapping to UML through BPMN Profiles; these profiles enable you to develop BPMN diagrams quickly and simply.	BPMN modeling ^[1545]
BPEL	Business Process Execution Language is an executable language for specifying interactions with Web Services. Enterprise Architect uses the BPMN profile as a graphical front-end to capture BPEL Process descriptions.	BPEL modeling ^[1570]
SPEM	The Software and Systems Process Engineering Meta-model (SPEM) is a conceptual framework for modeling, documenting, presenting, managing, interchanging, and enacting development methods and processes. SPEM 2.0 focuses on providing the additional information structures that you require for processes modeled with UML 2 Activities or BPMN/BPDM.	Software Process modeling (SPEM) ^[523]
ArchiMate	ArchiMate is an open-standard enterprise architecture language based on the IEEE 1471 standard, providing a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems and technical infrastructure.	ArchiMate ^[1621]

Topic	Detail	See also
	It enables Enterprise Architects to clearly describe, analyse and visualize the relationships among business domains.	
Data Flow Diagrams	<p>A data flow diagram (DFD) is a graphical representation of the flow of data through an information system, and can also be used to visualize data processing (structured design).</p> <p>Developing a DFD helps in identifying the transaction data in the data model.</p>	Data Flow Diagrams ^[1506]
Entity Relationship Diagrams	<p>Entity-relationship modeling is an abstract and conceptual database modeling method, used to produce a schema or semantic data model of, for example, a relational database and its requirements, visualized in Entity-Relationship Diagrams (ERDs).</p> <p>ERDs in Enterprise Architect assist you in building conceptual data models through to generating Data Definition Language (DDL) for the target DBMS.</p>	Entity Relationship Diagrams ^[1627]
Eriksson-Penker Extensions	<p>Eriksson-Penker extensions provide a framework for UML business processing model extensions, to which an Enterprise Architect can add stereotypes and properties appropriate to their business.</p> <p>In Enterprise Architect, the Eriksson-Penker profile provides, through a set of stereotypes, a unique and powerful means of visualizing and communicating business processes and the necessary flow of information within an organization.</p>	Eriksson-Penker Extensions ^[1623]
Mind Mapping	<p>A Mind Map is an image-centered diagram used to represent semantic or other connections between words, ideas, tasks or other items arranged radially around a central key word or idea.</p> <p>A Mind Map is used to generate, visualize, structure and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing.</p>	Mind Mapping ^[1503]
SoaML	Service Oriented Architecture (SOA) is an architectural paradigm for defining how people, organizations and systems provide and use services to achieve results.	SoaML ^[2088]
SOMF	The service-oriented modeling framework (SOMF) is a service-oriented development life cycle methodology, offering a number of modeling practices and disciplines that contribute to a successful service-oriented life cycle management and modeling.	SOMF ^[2092]

Topic	Detail	See also
ArcGIS	ArcGIS is a suite of Geographic Information Systems (GIS) software products developed by Esri. <i>Support for this product is under development, and in Beta stage.</i>	ArcGIS ^[1633]

8.3.1 Business Modeling/Interaction

Business Modeling diagrams and Business Interaction diagrams enable you to model both the structure and behavior of a business system. Business Modeling diagrams are based on a Class (UML Structural) diagram, whilst Business Interaction diagrams are based on a Sequence (UML Behavioral) diagram. Both diagram types have the same default Toolbox, which consists of a Business Modeling element page. The available elements include stereotyped Objects, and a stereotyped Actor (Business Actor), Use Case (Business Use Case) and Collaboration (Business Use Case Realization).

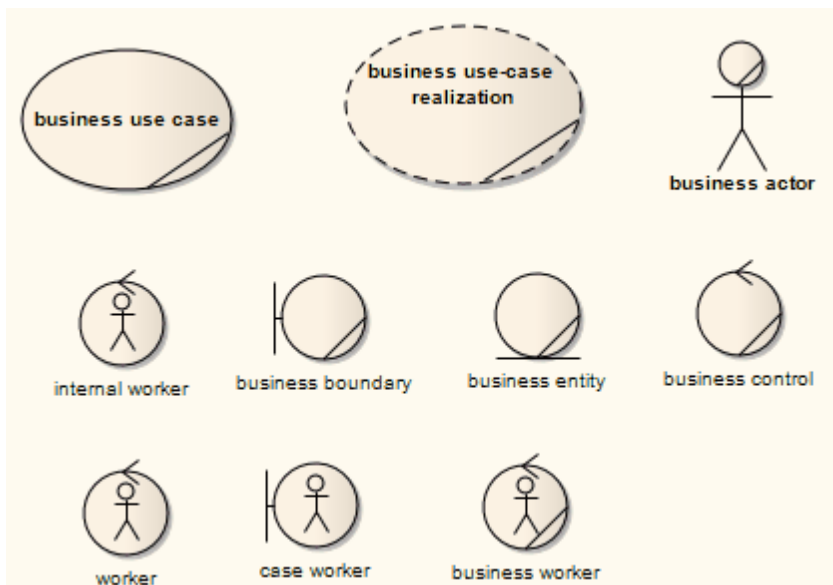
Example Diagram [Example Business Modeling Diagram](#) ^[1514]

Learning Center topics

- **(Alt + F1) | Modeling Languages | Business | Business Modeling and Interactions**

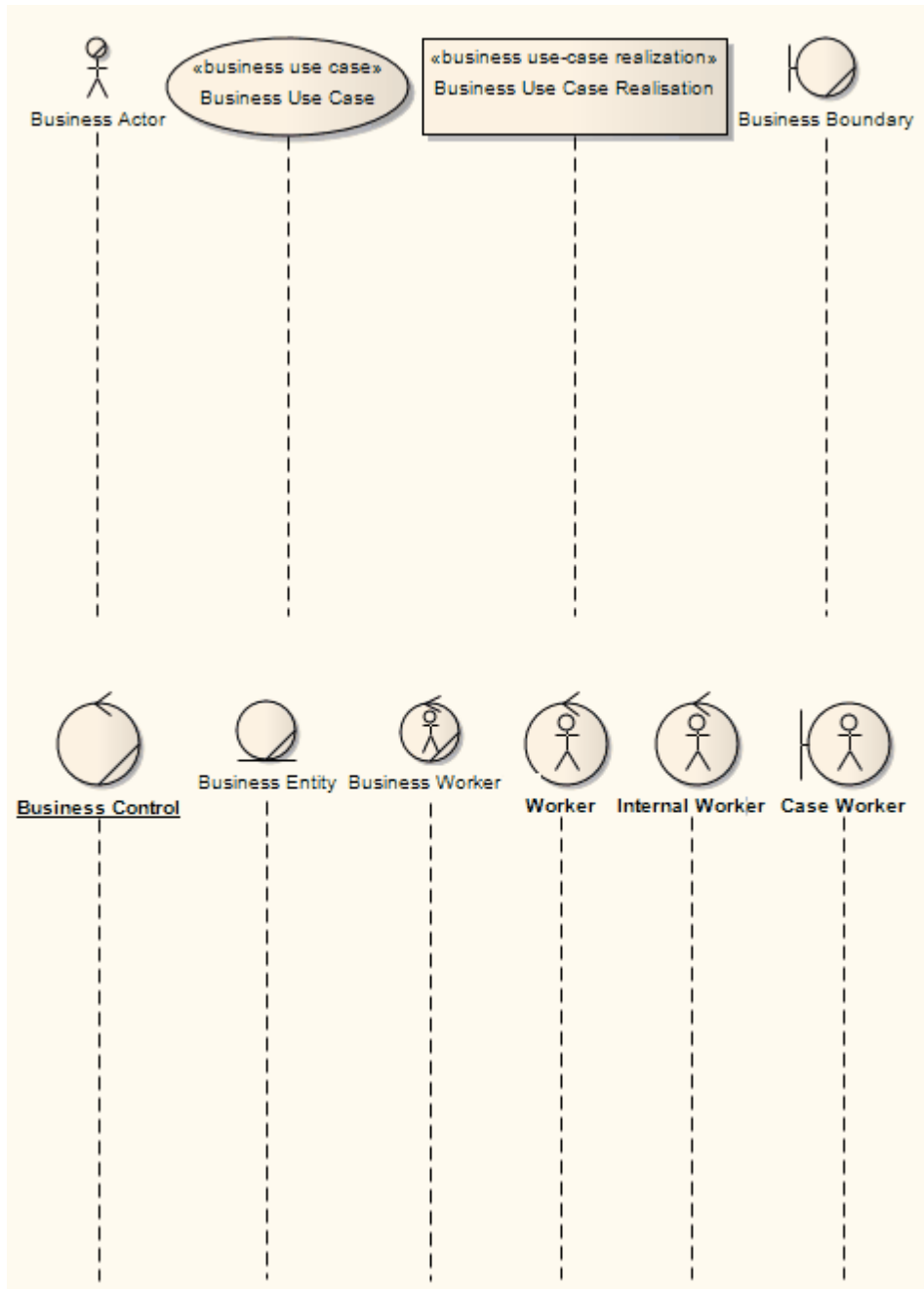
8.3.1.1 Example Business Modeling Diagram

The following diagram shows the appearance of the elements when dragged and dropped onto a Business Modeling diagram:



The following diagram shows the appearance of the elements when dragged and dropped onto a Business

Interaction diagram:



8.3.2 Business Models

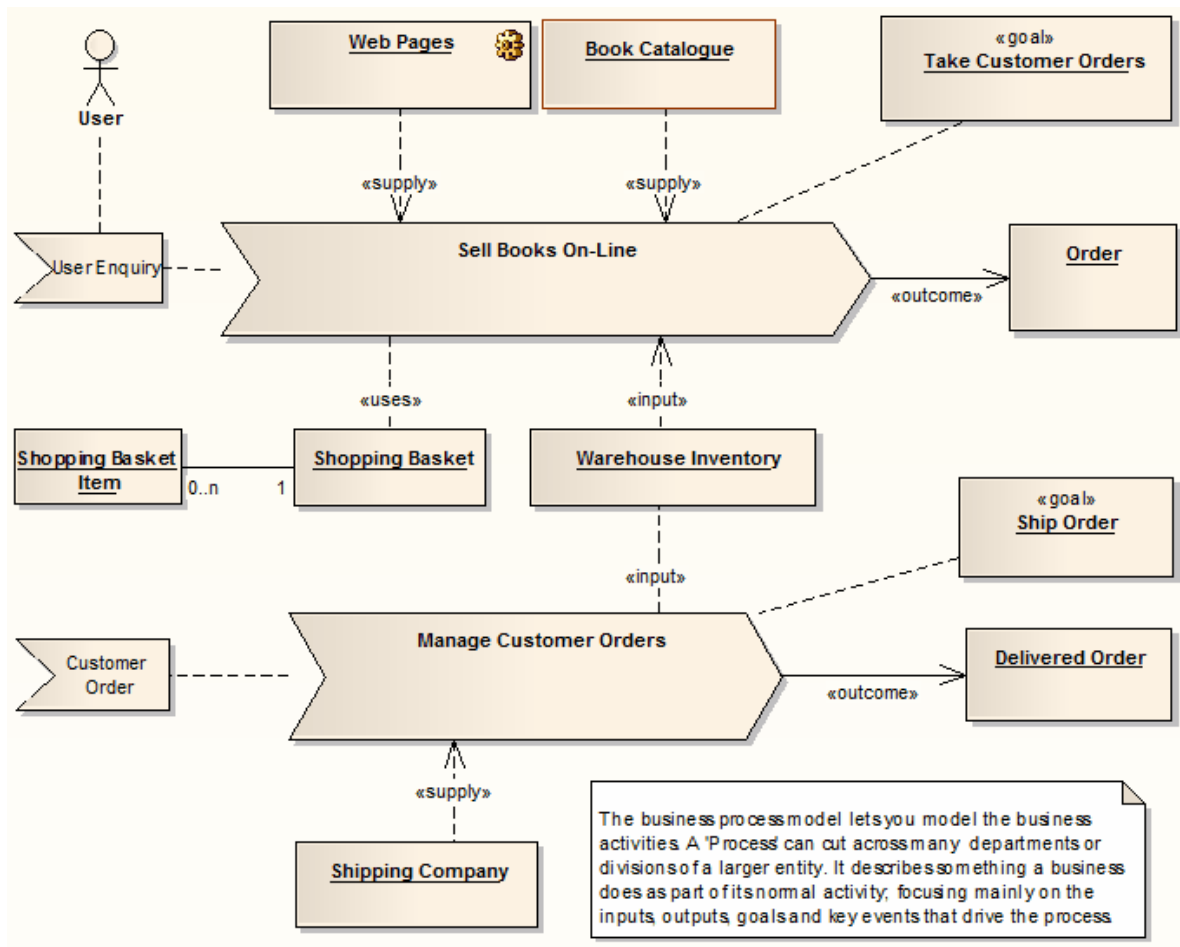
Topics

Topic	Detail	See also
Modeling the Business	Modeling the business process is an essential part of any software development process. It enables the analyst to capture the broad	Analysis Model 1517

Topic	Detail	See also
Process	<p>outline and procedures that govern what it is a business does. This analysis model provides an overview of where the proposed software system being considered fits into the organizational structure and daily activities. It can also provide the justification for building the system by capturing the current manual and automated procedures that are to be rolled up into a new system, and the associated cost benefit.</p> <p>As an early model of business activity, it enables the analyst to capture the significant events, inputs, resources and outputs associated with business process. By connecting later design elements (such as Use Cases) back to the business process model through Implementation connectors, it is possible to build up a fully traceable model from the broad process outlines to the functional requirements and eventually to the software artefacts actually being constructed.</p> <p>As the Business Process Model typically has a broader and more inclusive range than just the software system being considered, it also enables the analyst to clearly map what is in the scope of the proposed system and what is to be implemented in other ways (such as a manual process).</p>	

Example

The example below demonstrates the kind of model that can be built up to represent a business process. In this model, the goal of the business process is to take customer orders and to ship those orders out. A user starts the process with an inquiry, which leads to the involvement of the Book Catalogue, Shopping Cart, on-line pages and warehouse inventory. The output of significance to the business is a customer order.



The second half of the process model is to respond to a customer order and ship the required items. The second process involves the warehouse inventory and shipping company, and completes when an order is delivered to the customer.

Learn more

- [Business Modeling and Business Interaction Diagrams](#) ^[1514]
- [Web Stereotypes](#) ^[1660]

8.3.2.1 Analysis Models

This section discusses the development of analysis models to construct business processes. It describes:

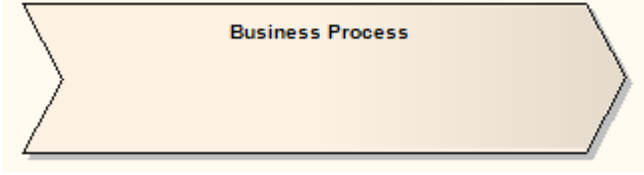
Topic	Link
Process Modeling Notation	Process Modeling Notation ^[1519]
Inputs, Resources and Information	Inputs, Resources and Information ^[1519]

Topic	Link
Events	Events <small>[1520]</small>
Outputs	Outputs <small>[1521]</small>
Goals	Goals <small>[1521]</small>
A Complete Business Process	A Complete Business Process <small>[1522]</small>

8.3.2.1.1 Process Modeling Notation

Topics

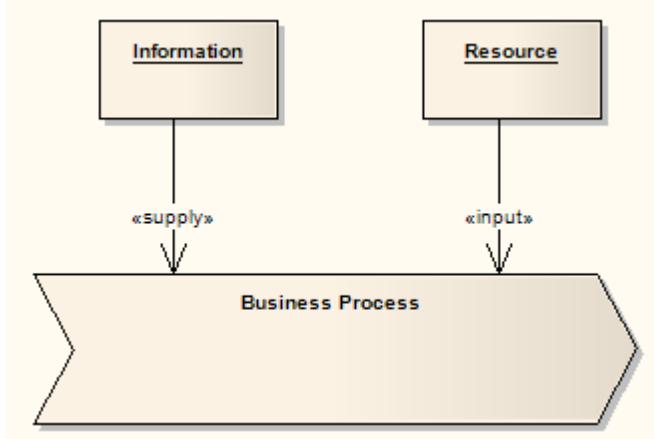
Topic	Detail	See also
Abstract	<p>A business process model typically defines the following elements:</p> <ul style="list-style-type: none"> • The goal or reason for the process • Specific inputs • Specific outputs • Resources consumed • Activities that are performed in some order, and • Events that drive the process <p>The business process:</p> <ul style="list-style-type: none"> • Can affect more than one organizational unit • Can have a horizontal organizational impact • Creates value of some kind for the customer; customers can be internal or external <p>A business process is a collection of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how the work is done within an organization, in contrast to a product's focus on what. A process is thus a specific ordering of work activities across time and place, with a beginning, an end, and clearly defined inputs and outputs: a structure for action. The notation used to depict a business process is illustrated below.</p>	<p>Process <small>[1674]</small></p> <p>Events <small>[1520]</small></p> <p>Activity <small>[1101]</small></p>

Topic	Detail	See also
	 <p>The process notation implies a flow of activities from left to right. Typically an Event element is placed to the left of the process and the output to the right. To specifically notate the internal activities, Activity elements can be placed inside the process element.</p>	
The BPMN File	<p>One popular notation and approach to business modeling is the Business Process Modeling Notation (BPMN). This notation is specifically targeted at the business modeling community and has a relatively direct mapping to UML through a BPMN Profile. Sparx Systems provides a built-in profile for BPMN modeling in Enterprise Architect.</p>	BPMN Models <small>[1545]</small>

8.3.2.1.2 Inputs, Resources and Information

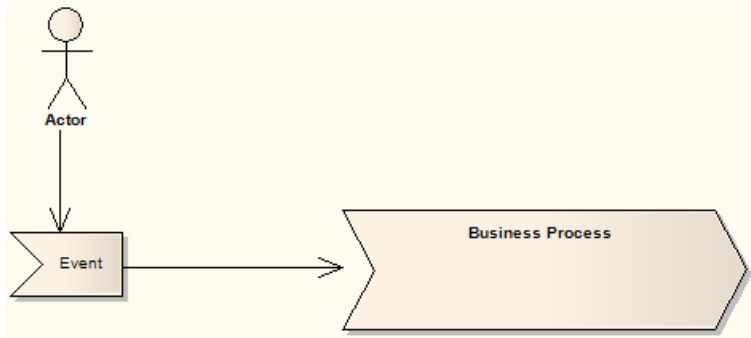
Topics

Topic	Detail	See also
Abstract	<p>Business processes use information to tailor or complete their activities. Information, unlike resources, is not consumed in the process; rather it is used as part of the transformation process. Information can come from external sources, from customers, from internal organizational units and could even be the product of other processes. A resource is an input to a business process and, unlike information, is typically consumed during the processing. For example, as each daily train service is run and actuals recorded, the service resource is 'used up' as far as the process of recording actual train times is concerned.</p> <p>The notation to illustrate information and resources is shown below.</p>	

Topic	Detail	See also
	 <p>A <i>Supply</i> connector indicates that the information or object linked to the process is not used up in the processing phase. For example, order templates can be used over and over to provide new orders of a certain style; the templates are not altered or exhausted as part of this activity.</p> <p>An <i>Input</i> connector indicates that the attached object or resource is consumed in the processing procedure. As an example, as customer orders are processed they are completed and signed off, and typically are used only once per unique resource (order).</p>	

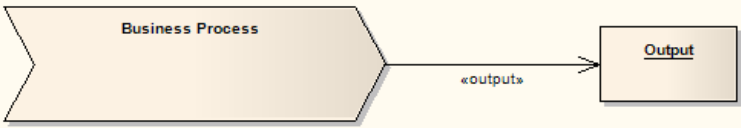
8.3.2.1.3 Events

Topics

Topic	Detail	See also
<p>Abstract</p>	<p>An event is the receipt of some object, a time or date reached, a notification or some other trigger that initiates the business process. The event might be consumed and transformed (for example a customer order) or simply act as a catalyst (for example, nightly batch job).</p> 	<p>Event^[1667]</p>

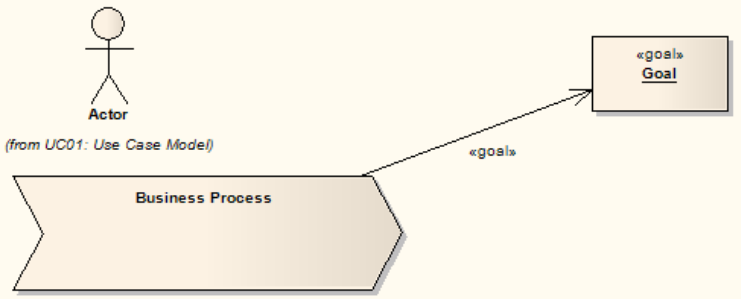
8.3.2.1.4 Outputs

Topics

Topic	Detail	See also
Abstract	<p>A business process typically produces one or more outputs of value to the business, either for internal use or to satisfy external requirements. An output might be a physical object (such as a report or invoice), a transformation of raw resources into a new arrangement (a daily schedule or roster) or an overall business result such as completing a customer order.</p> <p>An output of one business process might feed into another process, either as a requested item or a trigger to initiate new activities.</p>  <p>An Output connector indicates that the business process produces some object (either physical or logical) that is of value to the organization, either as an externally visible item or as an internal product (possibly feeding another process).</p>	

8.3.2.1.5 Goals

Topics

Topic	Detail	See also
Abstract	<p>A business process has some well defined goal. This is the reason the organization does this work, and should be defined in terms of the benefits this process has for the organization as a whole and in satisfying the business requirements.</p>  <p>A Goal connector indicates that the attached object to the business</p>	

Topic	Detail	See also
	process describes the goal of the process. A goal is the business justification for performing the activity.	

8.3.2.1.6 A Complete Business Process

Topics

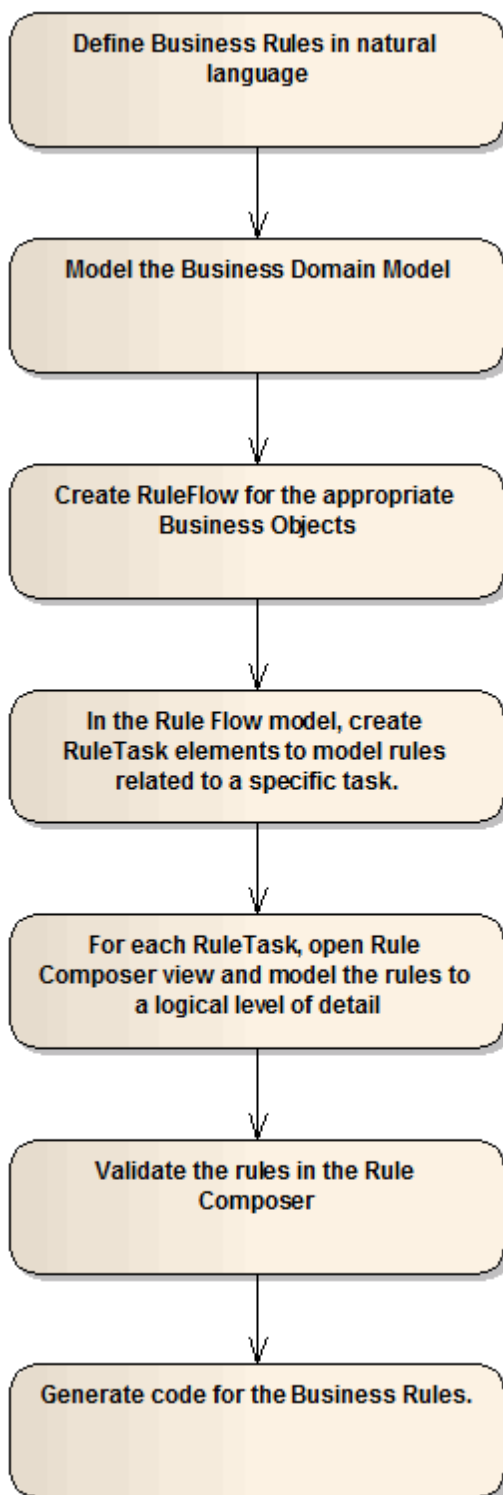
Topic	Detail	See also
Abstract	<p>The diagram below illustrates how the various model elements can be grouped together to produce a coherent picture of a named business process. Included are the inputs, outputs, events, goals and other resources that are of significance.</p>	

8.3.3 Business Rules

To model Business Rules in Enterprise Architect, work through the following steps

Example

These steps are represented graphically in the following flow:

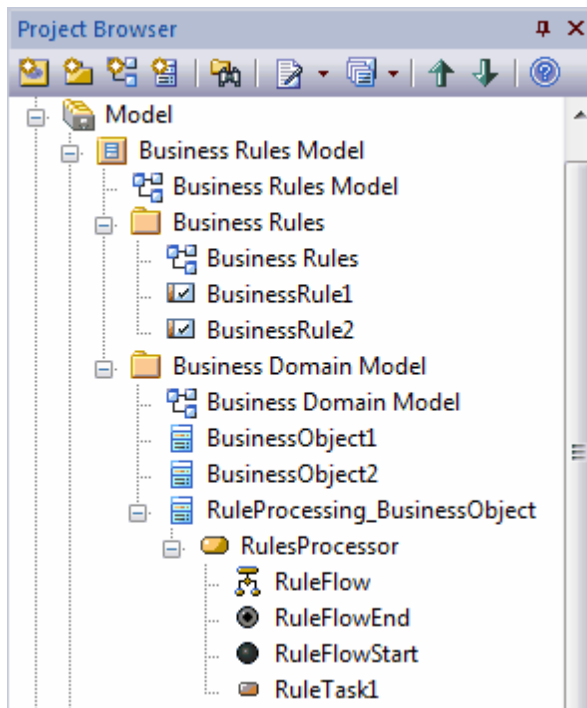
**How to**

To create a Business Rule Model from a template provided with Enterprise Architect

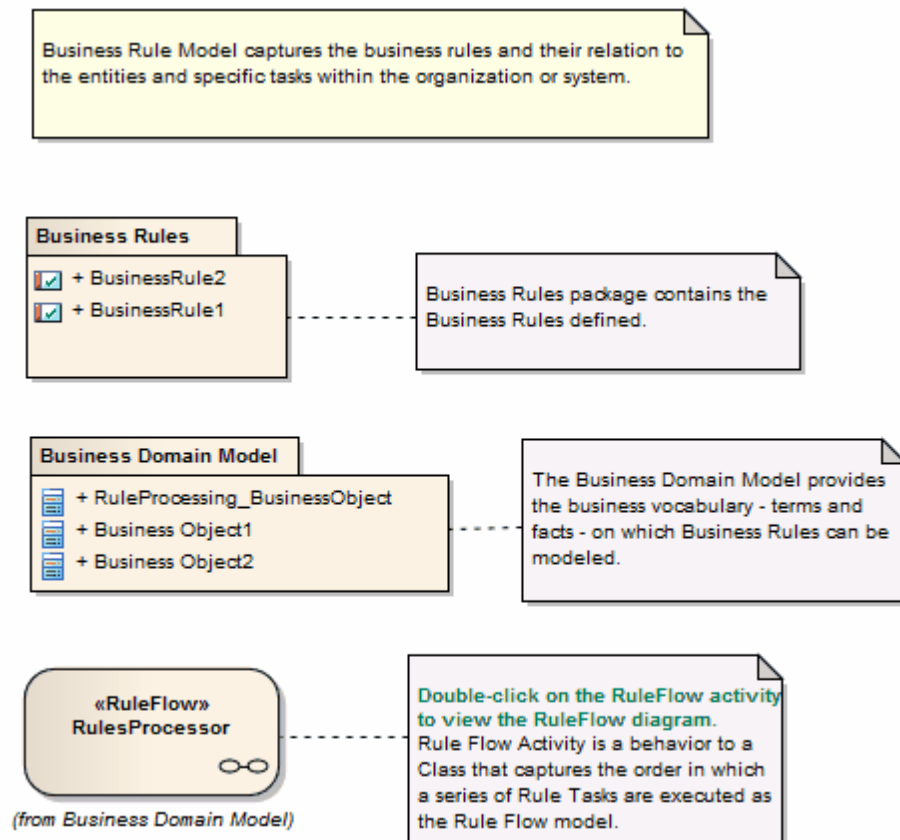
Step	Action	See also
1	<p>In the Project Browser, either:</p> <ul style="list-style-type: none">• Click on the New Model From Pattern icon in the toolbar• Right-click on a model root node and select the Add a New Model using Wizard context menu option• Right-click on a package and select the Add Add a New Model using Wizard context menu option. <p>The Select Model(s) dialog displays.</p>	
2	<p>In the Select From field, click on the drop-down arrow and select Business Rule Model.</p> <p>Alternatively, if it is listed in the Technology panel, select the Business Rule Model item.</p>	
3	<p>In the Name panel, select the checkbox next to the Business Rule Model icon.</p>	
4	<p>Click on the OK button.</p>	

The following model structure is created in the Project Browser:

Example



The *Business Rules Model* diagram, shown below, encapsulates the components of the Business Rules model.



Notes

- Business Rule Modeling is available in the Business and Software Engineering edition and Ultimate edition of Enterprise Architect

Learn more

- [Create a Rule model](#)^[1523]
- [Create a Business Domain Model](#)^[1529]
- [Create a Rule Flow Model](#)^[1530]
- [Compose Business Rules](#)^[1536]
- [Validate Business Rules](#)^[1542]
- [Code Generation for Business Rules](#)^[1543]

Learning Center topics

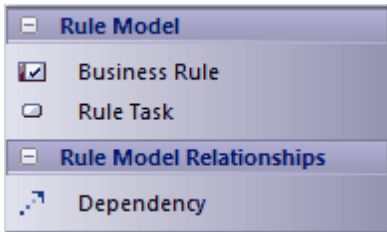
- **(Alt + F1) | Modeling Languages | Business | Business Rule Model Introduction**
- **(Alt + F1) | Business Modeling | Business Rules | Getting Started**

8.3.3.1 Model Business Rules For RuleTasks

The *Rule Model* enables you to define *Business Rule* elements and associate them with a Rule Task. In the example, you might define a set of rules to perform an eligibility check for a customer, to determine if the customer is eligible to rent a car.

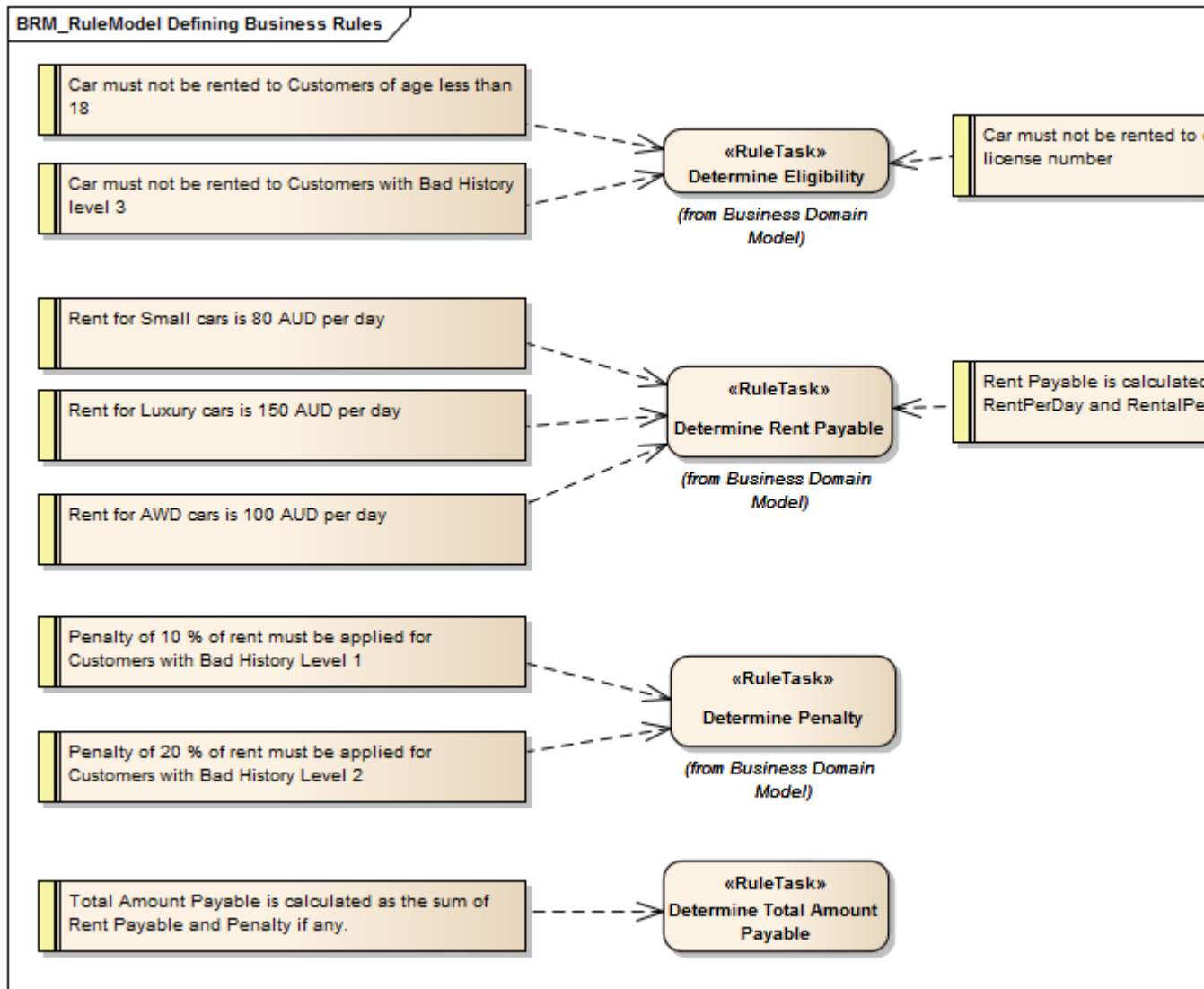
How to:

To define Business Rule elements and associate them with a Rule Task, follow the steps below:

Step	Action	See also
1	<p>Create a diagram of type <i>Rule Model</i>; the Rule Model pages display in the Toolbox.</p> 	
2	<p>Drag and drop a Rule Task element (<i>Determine Eligibility</i> in the example) from the Rule Flow Activity diagram package onto the Rule Model diagram.</p>	
3	<p>Drag as many Business Rule elements as necessary from the Toolbox (or Project Browser if they exist already) onto the diagram.</p> <p>You type the rule as the element name here, then define the parameters of the rules using the Rule Composer.</p>	Composing Business Rules <small>1536</small>
4	<p>Create a Dependency relationship between each Business Rule element and the Rule Task element.</p> <p>However, when you bring the rule into the Rule Composer, it automatically creates the Dependency relationship anyway.</p>	
5	<p>Repeat steps 2 - 4 for the next Rule Task element.</p>	

Example:

The resulting Rule Model resembles the following diagram:



After you have modeled rules for all the Rule Task elements in the Rule Flow diagram, the Business Domain model is ready for **code transformation**. The code templates for generating technology-specific rule code work hand-in-hand with the **EASL code templates** to generate the code for the Rule Flow diagram.

Learn more:

- [Code Generation for Business Rules](#) ^[1543]
- [EASL code templates](#) ^[1449]

Learning Center topics:

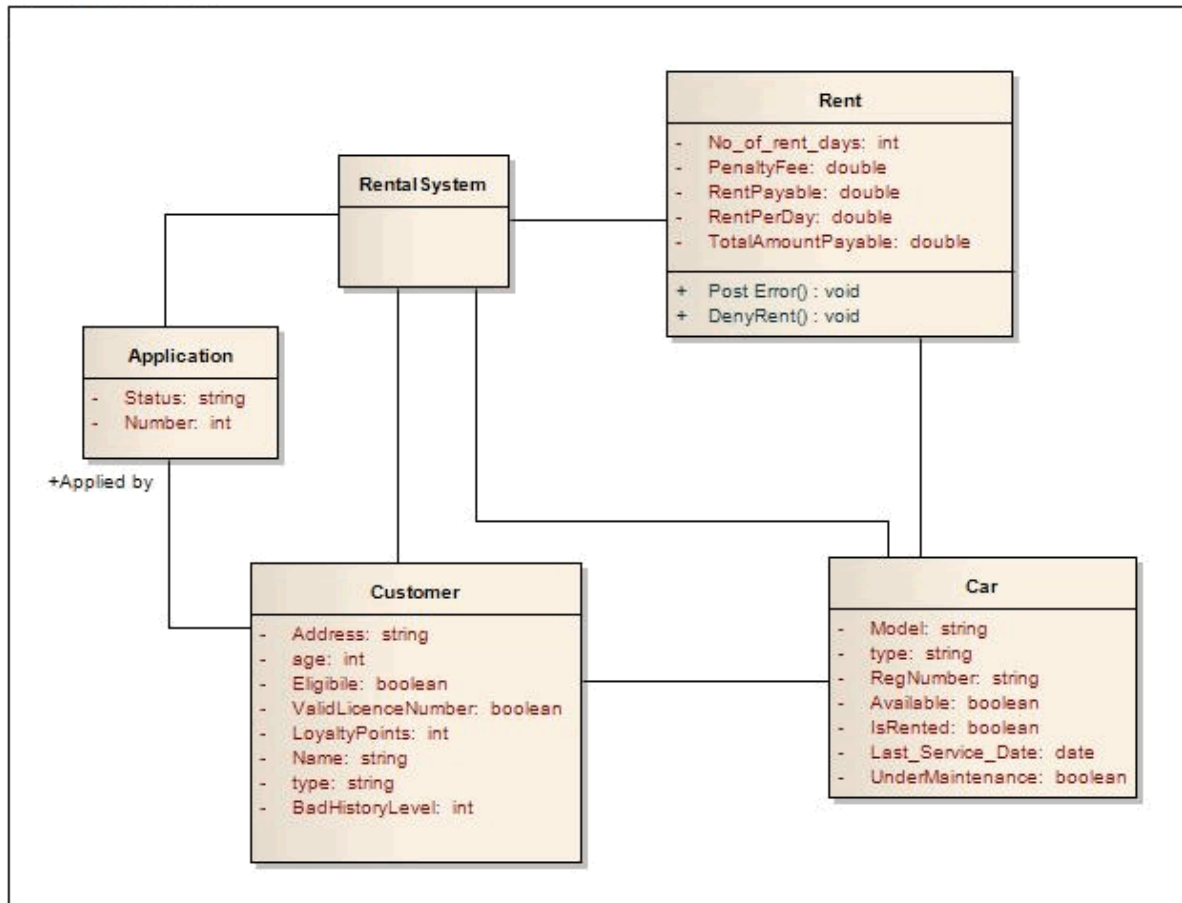
- **(Alt + F1) | Modeling Languages | Business | Business Rule Model**
- **(Alt + F1) | Business Modeling | Business Rules | Define Business Rules**

8.3.3.2 Create a Business Domain Model

The *Business Domain Model* provides the business vocabulary - terms and facts - on which Business Rules can be modeled. In Enterprise Architect a Business Domain model is created as a conceptual Class diagram.

Example

The following diagram shows an example Business Domain model, for a Car Rental system.



In the example Business Domain model, the Classes *Rent*, *Customer*, *Car* and *Application*, together with their attributes and operations, provide the terms for the business vocabulary. The Class *Rental System* processes the rules. To make *Rental System* process the rules, you add a Rule Flow Activity as a behavior for this Class.

When you create a Rule Flow behavior (Activity) under a Class you can model the rules as *Rule Tasks* (Actions). When code is generated the rule flow behavior is rendered as a method inside the corresponding Class.

Alternatively, if you have existing operations in the Class that already suit the purpose, you can model business rules in those operations. When code is generated for the Class the rules logic is generated as the method body for the corresponding operation.

Notes

- When you create Classes in the Business Domain model, select the correct language for code generation to ensure that the correct data type is set for attributes and operation parameters
- Business Rules code generation is supported for the following languages:
 - C++
 - C#
 - Java
 - VB.Net

Learn more

- [Create a Rule Flow Activity](#)^[1530]
- [Model Rules in an Operation](#)^[1534]

Learning Center topics

- **(Alt + F1) | Business Modeling | Business Rules | Business Domain Model**

8.3.3.3 Create a Rule Flow Model

You create a *Rule Flow Activity* as a behavior for a Class, to enable that Class to process a set of rules.

How to

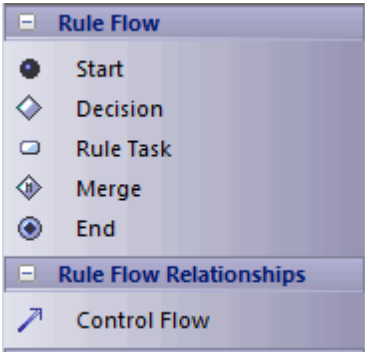
To add a Rule Flow Activity to a Class

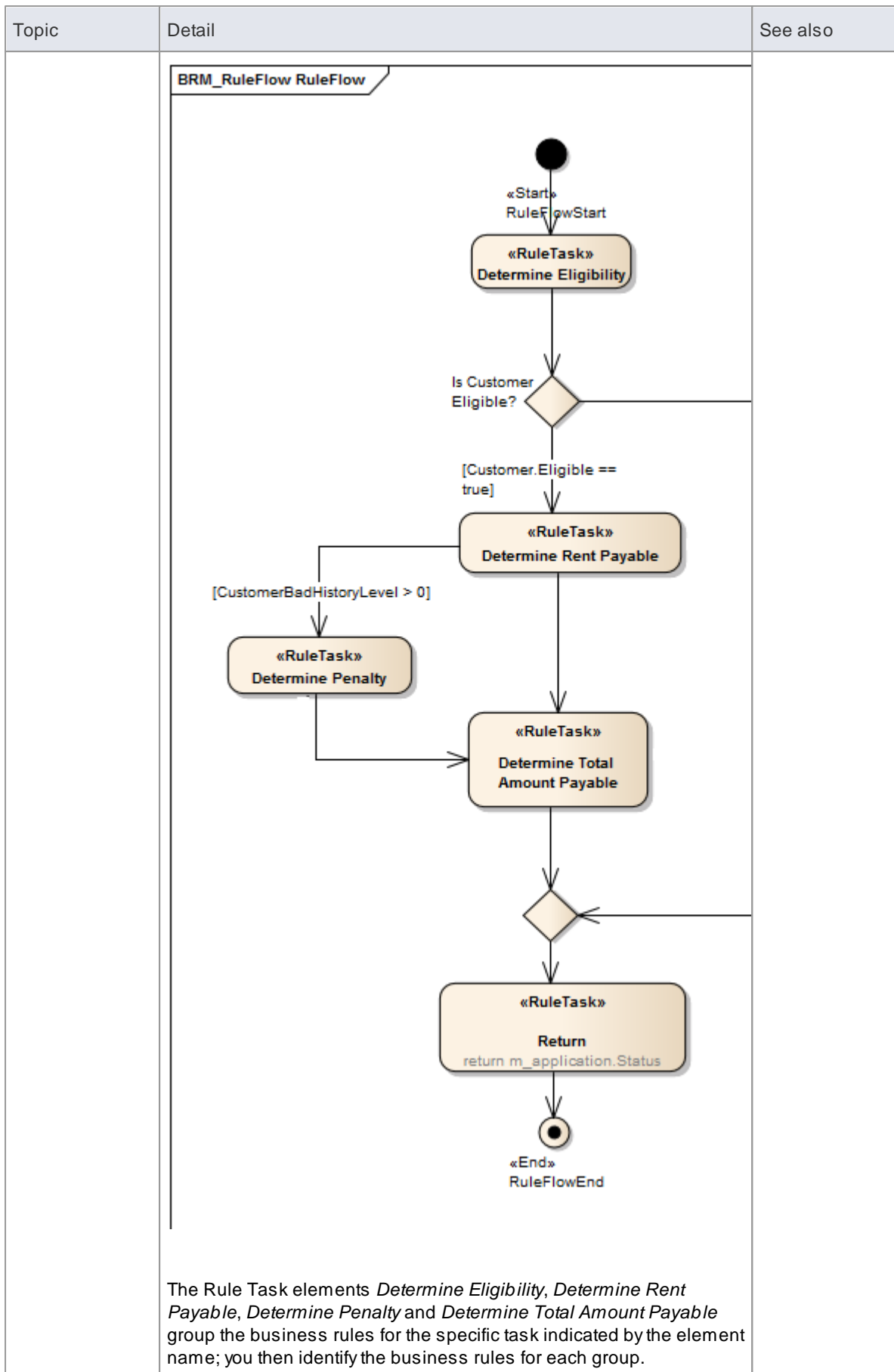
Step	Action	See also
1	On the Business Domain model diagram, right-click on the Class that processes the rules (in the example, <i>Rental System</i>).	
2	From the context menu select the Add RuleFlow Activity option.	

A new Rule Flow Activity with a *Rule Flow diagram* is created as a behavior for the selected Class. The Rule Flow diagram models the sequence in which a series of *Rule Tasks* are executed.

Code generation for a Rule Flow model renders each RuleFlow Activity as a set of operations or methods. Depending on what you want these methods to do, you might want to pass in some parameters to be used within the Rule Flow Activity. See the *Pass Parameters to Rule Flow Activity* topic.

Topics

Topic	Detail	See also
Add a Rule Task	<p>A Rule Task is a stereotyped Action that groups Business Rules for a specific task.</p> <p>You create Rule Task elements in a Rule Flow diagram using the associated Rule Flow pages of the Toolbox.</p>  <p>The following illustration is of a possible Rule Flow diagram for the car rental example.</p>	Model Business Rules ^[1527]



Topic	Detail	See also

Notes

- In a Rule Flow diagram, every Decision Node has a matching *Merge Node* to ensure proper code generation
- For code generation, the Rule Task elements must be grouped inside the appropriate Rule Flow Activity in the Project Browser; however, Rule elements can be defined anywhere in the model, as they can be used in more than one Rule Task

Learn more

- [Pass Parameters to Rule Flow Activity](#)^[1533]

Learning Center topics

- (Alt + F1) | **Modeling Languages | Business | Business Rule Model Flow**
- (Alt + F1) | **Business Modeling | Business Rules | Rule Flow Model**

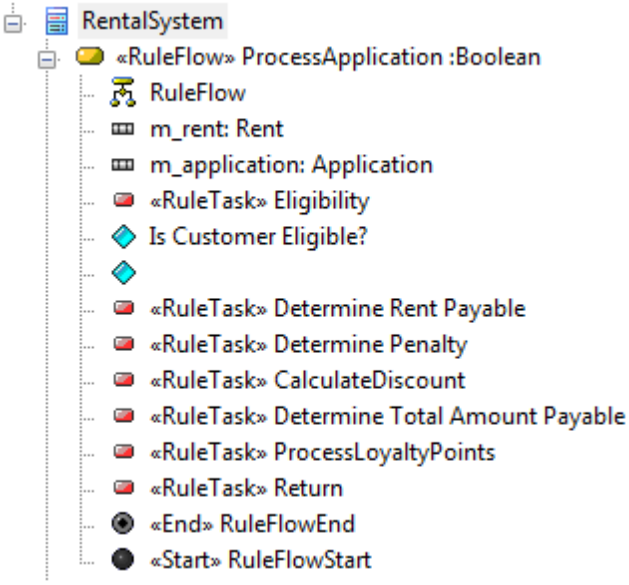
8.3.3.3.1 Pass Parameters to Rule Flow Activity

How to

To pass in parameters to be used within a Rule Flow Activity

Step	Action	See also
1	In the Project Browser, double-click on the Rule Flow Activity. The element Properties dialog displays.	
2	Click on the Behavior tab.	
3	Click on the Edit Parameters button. The Parameters dialog displays.	
4	Create and define each parameter, setting Type and Default values.	Parameters Dialog ^[913]
5	Save each parameter and, when you have finished setting the parameters, close both dialogs.	

Topics

Topic	Detail	See also
<p>Rule Flow Activity Parameters</p>	<p>The Rule Flow Activity parameters can be accessed by the Rule Tasks within the parent Rule Flow Activity.</p> <p>In the following hierarchy, the parameters <i>m_rent</i> and <i>m_application</i> can be used by any of the Rule Tasks under the <i>ProcessApplication</i> Rule Flow Activity.</p>  <p>You can use the parameters as condition variables or action variables in the Business Rule Decision Table, or as rule variables in the Computation Table for any of the Rule Tasks.</p> <p>If the Activity parameter is not accessible to a Rule Task, Enterprise Architect displays an error message.</p>	<p>Compose Business Rules</p>

8.3.3.3.2 Model Rules In an Operation

You can model business rules either in the Business Rule elements attached to the Rule Task element in a RuleFlow Activity diagram, or in the operations of the rule Class in the Business Domain model.

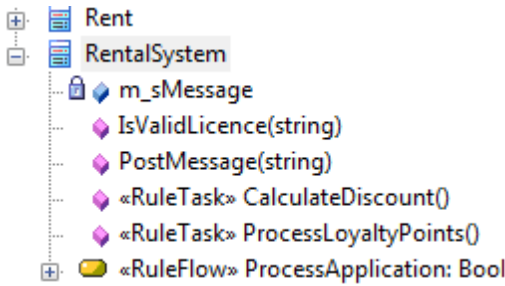
How to

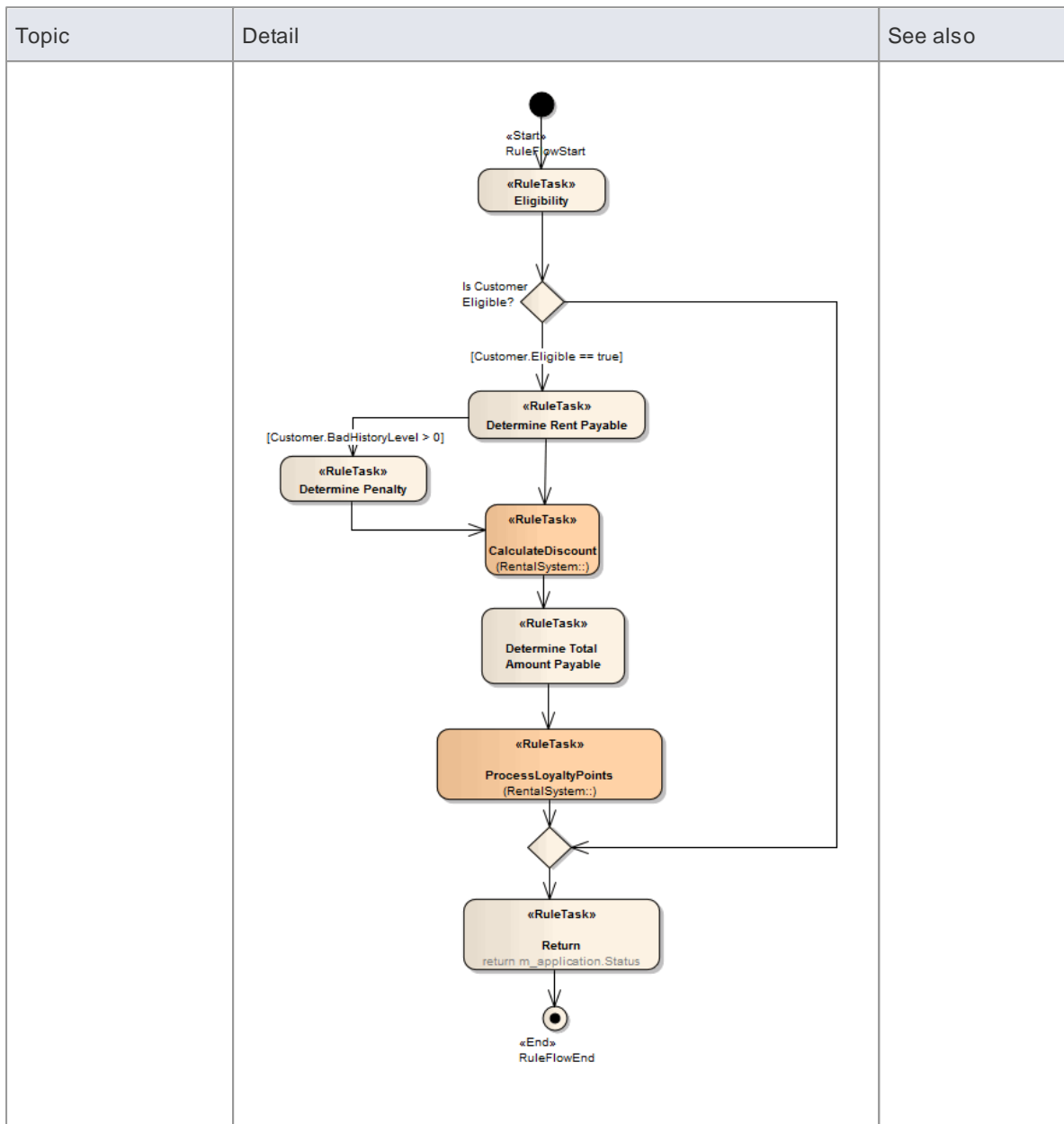
To model business rules for an operation

Step	Action	See also
1	Open the Properties dialog for the operation and, in the Stereotype field on the General tab, type the value RuleTask .	

Step	Action	See also
2	In the Project Browser, right-click on the operation and select the Rule Composer context menu option to open the Rule Composer.	
3	Model the rules for the operation.	Compose Business Rules [1536]

Topics

Topic	Detail	See also
Usage	<p>The operations appear in the Project Browser as shown below:</p>  <p>On code generation, the code for rules logic is generated in the method body.</p> <p>When you drag and drop a RuleTask operation onto a Rule Flow diagram, an operation call behavior action is created.</p> <p>To pass the parameters for this operation call, open the Properties dialog and select the Call tab, then set the Behavior field to the operation to be called; under the Arguments field, click on the Edit Arguments button and edit the argument values to be passed.</p> <p>On the diagram, the call behavior actions for the RuleTask operations are shown as illustrated below:</p>	<p>Behaviour Calls [910]</p> <p>Behaviour Call Arguments [912]</p>



Learn more

- [Business Rules](#)^[1527]
- [RuleFlow Activity Diagram](#)^[1530]
- [Business Domain Model](#)^[1529]

8.3.3.4 Compose Business Rules

You use the Rule Composer to define a business rule written in plain text within a Business Rule element or Class operation. The Rule Composer enables you to model conceptual-level business rules at a logical level in tabulated format, which assists in transforming the rules to technology-specific rules (code).

You can also download the contents of the Rule Composer to a spreadsheet application such as Microsoft Excel, via a CSV file.

Topics

Topic	Detail	See also
Access The Rule Composer	<p>To access the Rule Composer, right-click on a Rule Task element and select the Rule Composer context menu option; the Rule Composer displays in the central work area on its own tab.</p> <p>The Rule Composer consists of:</p> <ul style="list-style-type: none"> • a Rule Statements list • a Decision Table and • a Computation Rule Table. <p>To assist with traceability, as the Rule Composer is completed, selections in one table automatically highlight the corresponding rows and columns of the other tables; for example, if a Rule Statement is selected, the related rule column in the Decision Table and row in the Computation Rule Table are highlighted.</p> <p>Similarly, if a Computational Rule is selected, the corresponding column in the Decision Table and row in the Rule Statements list are highlighted.</p>	
Rule Statements Table	<p>The Rule Statements table lists the rules associated with the selected Rule Task; you add a rule to the table by dragging an existing Business Rule element from the Project Browser onto an empty row in the Rule Statements table.</p> <p>You cannot create new rules within the table.</p>	
Decision Table	<p>The Decision Table enables you to model <i>conditional</i> rules (for example: <i>Cars must not be rented to customers of age less than 18</i>)</p> <p>The table has three sections:</p> <ul style="list-style-type: none"> • Rule Conditions – to model condition variables • Rule Actions – to model action variables • Rule Bindings – to link the rule in the rule table. 	
Computation Rule Table	<p>The Computation Rule table enables you to model rules involving computations</p> <p>The table has the following columns:</p> <ul style="list-style-type: none"> • Computation Rule Actions • Expression • Rule Bindings • Rule Dependency. 	

How to

To define a business rule associated with the selected Rule Task

Step	Action	See also
1	For the first rule, select the text within the Business Rule element and drag it onto the empty row.	
2	For a subsequent rule, click on the No column and select the Add Row context menu option; an empty row is added to the Rule Statements table.	
3	Drag the required Business Rule element from the Project Browser onto the new row. If the Business Rule element is not already on the diagram, this adds the element to the diagram and creates a Dependency relationship between the Business Rule and Rule Task elements.	

To remove a rule that is no longer required in the Rule Composer, right-click on the appropriate **No** field and select the **Remove Rule** context menu option. This removes the rule from the Rule Composer and deletes the Dependency relationship with the Rule Task element. However, it does not remove the Business Rule element from either the diagram or the Project Browser (where, in either case, it might be in use with other Rule Task elements).

To model Rule Conditions in the Rule Conditions Section

Step	Action	See also
1	The Business Domain model defines the business terms (such as <i>Customer</i>) and their associated attributes; from the appropriate Class element in the Project Browser, drag and drop the required condition attribute (such as age) or operation (such as <i>IsValidLicense()</i>) onto the Rule Conditions column. <ul style="list-style-type: none"> The Rule Condition field enables you to use intellisense to display a list of possible entries for the field; press (Ctrl+Spacebar) in the field to display the list of entries If the Rule Condition is of type <i>enum</i>, the Allowable Values fields are automatically set with the <i>enum</i> literals; the procedure then ends here. 	
2	Determine a range of accepted values for the Rule Condition.	
3	Right-click on the Allowable Values column and select the Edit Allowable Values context menu option; the Edit Allowable Values dialog displays. Type each required value or range of values in the Value field, and click on the Save button to display the value in the Allowable Values list box	

Step	Action	See also
	<p><i>age</i> could have the values:</p> <p><18 >18 and <50 >50</p> <p><i>IsValidLicense()</i> could return:</p> <p>True False</p>	
4	<p>Click on the OK button to save the values and close the dialog; a new constraint <i>Allowable Values</i> is created for the attribute.</p> <ul style="list-style-type: none"> You can check this constraint by opening the Properties dialog for the attribute and selecting the Constraints tab If the Rule Condition references an enumeration, the enum literals are not editable in the Edit Allowable Values dialog. 	
5	<p>If the Rule Condition is an operation, you can pass parameters to it.</p> <p>Right click on the Allowable Values field, and select the Edit Parameters context menu option; the Edit Parameters dialog displays.</p> <p>Select the parameters and type their values into the Value text box; click on the OK button to cancel the dialog.</p> <ul style="list-style-type: none"> You can add an operation as a Rule Condition more than once, to allow calling the operation with different sets of parameters To add another Rule Condition, right-click on the No column and select the Add Row context menu option; an empty row is added to the table. <p>To remove a Rule Condition from the table, right-click on the appropriate No field and select the Delete Row context menu option; this does not affect the original attribute or <i>the new constraint</i> in the model.</p> <p>You can either re-use the attribute with its constraint, or use the attribute Properties dialog to remove the constraint.</p>	

To model Rule Actions in the Rule Actions section

Step	Action	See also
1	<p>In the Rule Actions section, when a specific value of a Rule Condition calls an operation (such as <i>post error</i>) or decision attribute (such as <i>Eligible - Yes/No</i>), you assign the operation or attribute as an action.</p> <p>From a business term Class element in the Project Browser, drag and drop the required attribute or operation onto the Rule Actions field.</p>	

Step	Action	See also
	The Rule Actions field enables you to use intellisense to display a list of possible entries for the field; press (Ctrl+Spacebar) in the field to display the list of entries.	
2	<p>For an attribute, double-click on the Allowable Values/Parameters field; the Edit Allowable Values dialog displays.</p> <p>Type the range of values in the text box (such as Yes, No; or Accept, Reject), click on the Save button and close the dialog; select the appropriate response in the Result column fields.</p> <p>If the dropped action variable is of type <i>enum</i>, the Allowable Values/Parameters fields are automatically set with the enum literals.</p>	
3	<p>For an operation, a checkbox displays in each of the Result column fields; to call the operation, select the checkbox in the appropriate column.</p> <p>To pass parameters to the operation, double-click on the Allowable Values/Parameters field; the Edit Parameters dialog displays.</p> <p>Select the parameters and type the values into the Value text box; click on the Save button and close the dialog.</p> <p>You can add an operation as a Rule Action more than once, to allow calling the operation with different sets of parameters.</p>	
4	<p>Click on the Save button in the Rule Composer toolbar to save the values.</p> <p>Alternatively, you can <i>right-click</i> on an Allowable Values/Parameters field to display a context menu with two options:</p> <ul style="list-style-type: none"> • If the Rule Action is an attribute, the Edit Allowable Values option is enabled and this displays the Edit Allowable Values dialog • If the Rule Action is an operation, the Edit Parameters option is enabled and this displays the Edit Parameters dialog <p>To add another Rule Action, right-click on the No column and select the Add Row context menu option; an empty row is added to the table.</p> <p>To remove a Rule Action from the table, right-click on the appropriate No field and select the Delete Row context menu option; this does not affect the original attribute or operation in the model.</p>	

To bind a rule in the Rule Bindings Section


Step	Action	See also
1	<p>The Rule Bindings section lies on top of the Rule Conditions section; it binds the Rule Condition and Rule Action values to the appropriate rule in the Rule Table.</p> <p>Select the rule number in the Rule Bindings field over one of the Value<n> or</p>	

Step	Action	See also
	Result<n> columns.	
2	Ensure that the values set in the Value<n> or Result<n> field for the Rule Condition or Rule Action, underneath the rule number, all satisfy the rule.	
3	<p>Click on the Save icon in the Rule Composer toolbar.</p> <p>For example, if rule 2 is <i>Car must not be rented to Customers of age less than 18</i>:</p> <ul style="list-style-type: none"> • Select 2 in the Rule field over the Value1 column • Select < 18 against <i>Customer.age</i> in the Value1 column in the Rule Conditions table • Select No against <i>Customer.Eligible</i> in the Result1 column in the Rule Action table • Select Reject against <i>Application.Status</i> in the Result1 column in the Rule Action table. 	

To define a computation rule

Step	Action	See also
1	<p>From the Project Browser, drag and drop the appropriate attribute from a Class in the <i>Fact</i> model into the Computation Rule Actions field.</p> <p>Both the Computation Rule Actions field and the Expression field enable you to use intellisense to display a list of possible entries for the fields; press (Ctrl + Spacebar) in the field to display the list of entries.</p>	
2	In the Expression field, type the expression to be evaluated.	
3	In the Rule Bindings field, type the rule number from the Rule table of the rule being modeled, to link the table data to the rule.	
4	If the rule depends on another rule being satisfied first, type the number of that rule in the Rule Dependency field.	
5	<p>Click on the Save icon in the Rule Composer toolbar to save the computation rule.</p> <p>If the computation rule is also a Rule Conditions rule, add the condition variable in the Decision table and bind the appropriate rule in the Rule Bind section.</p>	

To export the contents of the Rule Composer to a CSV file, follow the steps below

Step	Action	See also
1	Click on the Export to CSV icon () in the Rule Composer toolbar; the Windows Browser dialog displays.	
2	Browse to the required file location and type in a .CSV filename to export to.	
3	Click on the Save button to export the data.	

Learn more


- [Code Generation for Business Rules](#)^[1543]
- [CSV Import and Export](#)^[435]

Learning Center topics

- (Alt + F1) | **Business Modeling | Business Rules | Rule Composer**

8.3.3.5 Validate Business Rules

Topics

Topic	Detail	See also
Usage	<p>It is recommended practice to validate the business rules in the Rule Composer before you generate code for the Rule Task elements; to do this, click on the Validation (green tick) icon in the Rule Composer toolbar:</p>  <p>The business rules on the Rule Composer are parsed and any errors or warnings that might indicate incomplete or unfavorable code generation are displayed on a Rule Composer Validation tab on the Output screen.</p> <p>To highlight and investigate the faulty data in the Rule Composer, double-click on the appropriate warning or error message.</p>	

Learning Center topics

- (Alt + F1) | **Business Modeling | Business Rules | Validate Business Rules**

8.3.3.6 Code Generation For Business Rules

After you have modeled the business rules for all the Rule Task elements in the Rule Flow diagram, you can generate code from the Rule Flow behavior.

How to

To return a value from the Rule Flow behavior

Step	Action	See also
1	Double-click on the last Rule Task element before the end node of the Rule Flow diagram. The element's Properties dialog displays.	
2	Click on the Effect tab.	
3	In the Effect field, type the <i>return</i> statement; for example, <i>return true</i> .	
4	Click on the Save button, and on the OK button to close the dialog. Generate code for the Class containing the rule flow behavior (in our initial example, <i>Rental System</i>); the code for business rules logic is generated, with the rule statements expressed in natural language as comments.	Generate code for a single class ^[1770] Create a Business Domain Model ^[1529]

Example

The following code snippet was generated from the *Rental System* Class element in our example:

```

////////////////////////////////////
// Rental System.cs
// Implementation of the Class Rental System
// Generated by Enterprise Architect
// Created on:      08-May-2009 2:39:23 PM
////////////////////////////////////

public class Rental System {

    public Customer m_Customer;
    public Car m_Car;
    public Rent m_Rent;

    public Rental System() {

    }

    ~Rental System() {

    }

    public virtual void Dispose() {

```

```

}

/* Begin - EA generated code for Activities and Interactions */
public bool ProcessApplication(Rent m_rent, Application m_application)
{
    // behavior is an Activity

    /* CAR MUST NOT BE RENTED TO CUSTOMERS WITHOUT A VALID
LICENCE NUMBER*/
    if( m_Customer.ValidLicenceNumber == "FALSE" )
    {
        m_application.Status = "Reject";
        m_Customer.Eligible = false;
    }
    /* CAR MUST NOT BE RENTED TO CUSTOMERS OF AGE LESS THAN
18*/
    if( m_Customer.age < 18 )
    {
        m_application.Status = "Reject";
        m_Customer.Eligible = false;
    }
    /* CAR MUST NOT BE RENTED TO CUSTOMERS WITH BAD HISTORY
LEVEL 3*/
    if( m_Customer.BadHistoryLevel == 3 )
    {
        m_application.Status = "Reject";
        m_Customer.Eligible = false;
    }
    if ( Customer.Eligible == true)
    {

        /* RENT FOR SMALL CARS IS 80 AUD PER DAY*/
        if( m_Car.type == Small )
        {
            m_rent.RentPerDay = 80;
        }
        /* RENT FOR AWD CARS IS 100 AUD PER DAY*/
        if( m_Car.type == AWD )
        {
            m_rent.RentPerDay = 100;
        }
        /* RENT FOR LUXURY CARS IS 150 AUD PER DAY*/
        if( m_Car.type == Luxury )
        {
            m_rent.RentPerDay = 150;
        }
        /* RENT PAYABLE IS CALCULATED AS THE PRODUCT OF
RENTPERDAY AND RENTALPERIOD IN DAYS*/
        m_rent.RentPayable = m_rent.RentPerDay * m_rent.
No_of_rent_days;
        if ( Customer.BadHistoryLevel > 0)
        {

            /* PENALTY OF 20 % OF RENT MUST BE APPLIED FOR
CUSTOMERS WITH BAD HISTORY LEVEL 2*/
            if( m_Customer.BadHistoryLevel == 2 )
            {
                m_rent.PenaltyFee = m_rent.RentPayable
* 0.2;
            }
            /* PENALTY OF 10 % OF RENT MUST BE APPLIED FOR
CUSTOMERS WITH BAD HISTORY LEVEL 1*/
            if( m_Customer.BadHistoryLevel == 1 )
            {
                m_rent.PenaltyFee = m_rent.RentPayable
* 0.1;
            }
        }
    }
    else

```

```

        {
        }

        /* TOTAL AMOUNT PAYABLE IS CALCULATED AS THE SUM OF
        RENT PAYABLE AND PENALTY IF ANY. */
        m_rent.TotalAmountPayable = m_rent.RentPerDay +
        m_rent.PenaltyFee;
    }
    else
    {
    }
    return m_application.Status;
}

/* End - EA generated code for Activities and Interactions */
} // end Rental System

```

8.3.4 BPMN Models

The BPMN notation is specifically targeted at the business modeling community and has a direct mapping to UML through BPMN Profiles integrated with the Enterprise Architect installer. Through use of these profiles, Enterprise Architect enables you to develop BPMN diagrams quickly and simply.

The Enterprise Architect installer for releases later than 8.0 provides you with separate versions of the MDG Technology for BPMN that support BPMN versions 1.0, 1.1 and 2.0, and BPEL.

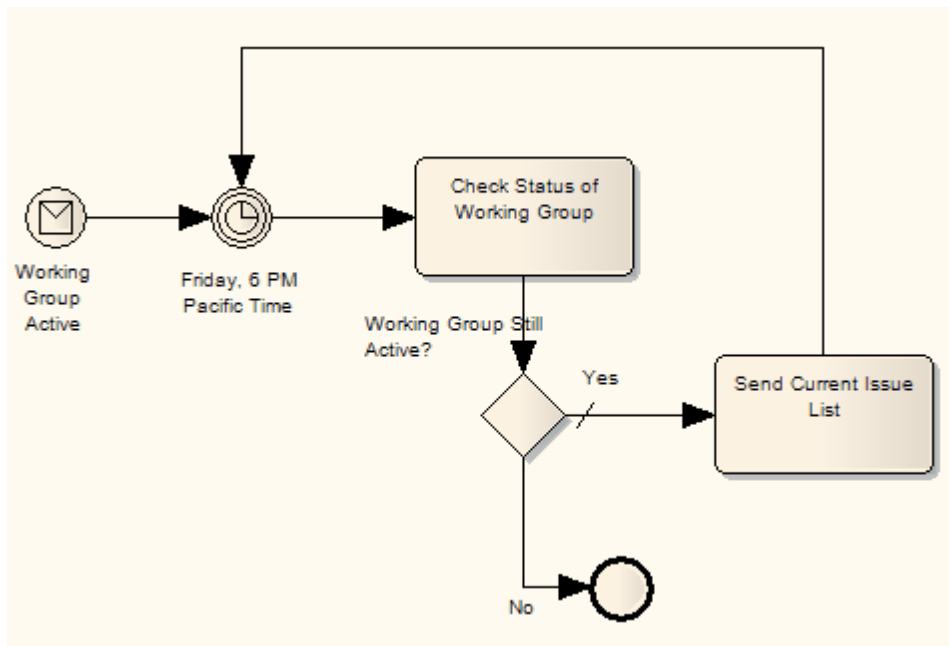
Use to

- Maintain existing diagrams created in BPMN 1.0 format, and create new diagrams in BPMN 1.0
- Create and maintain diagrams in the BPMN 1.1 and BPEL formats
- Create and maintain diagrams in the BPMN 2.0 and BPEL formats
- Migrate a BPMN 1.0 model (or part of a model) to BPMN 1.1

Access The BPMN facilities are provided in the form of:

- A BPMN diagram type, accessed through the New Diagram dialog
- BPMN pages in the Toolbox
- BPMN element and relationship entries in the Toolbox **Shortcut** menu and Quick Linker.

Specifications of BPMN elements and relationships are defined by Tagged Values (for example, to define the *Message*, *Timer* and *Default Path (/)* symbols in the diagram below).



BPMN Toolbox Pages:

You can access the BPMN Toolbox pages through the **View | Diagram Toolbox: More tools...** option; select the **BPMN 1.0, 1.1** or **2.0** options as appropriate.

You can also set BPMN as the active default technology to access the Toolbox pages directly.

Disable BPMN:

If you prefer not to use BPMN in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (**Settings | MDG Technologies**).

Learn more

- [BPMN 1.0 and 1.1 Toolbox Pages](#) ^[1547]
- [BPMN 2.0 Toolbox Pages](#) ^[1549]
- [BPMN 2.0 XML](#) ^[1569]
- [Migrate a BPMN 1.0 Model to BPMN 1.1](#) ^[1565]
- [Migrate a BPMN 1.1 Model to BPMN 2.0](#) ^[1567]
- [Change BPMN Element Appearance](#) ^[1561]
- [BPEL Models](#) ^[1570]
- [New Diagram dialog](#) ^[724]
- [Set Technology as Active Default](#) ^[1266]

Learning Center topics

- **(Alt + F1) | Modeling Languages | Business | BPMN | Introduction**

- (Alt + F1) | [Business Modeling](#) | [BPMN](#) | [Getting Started](#)

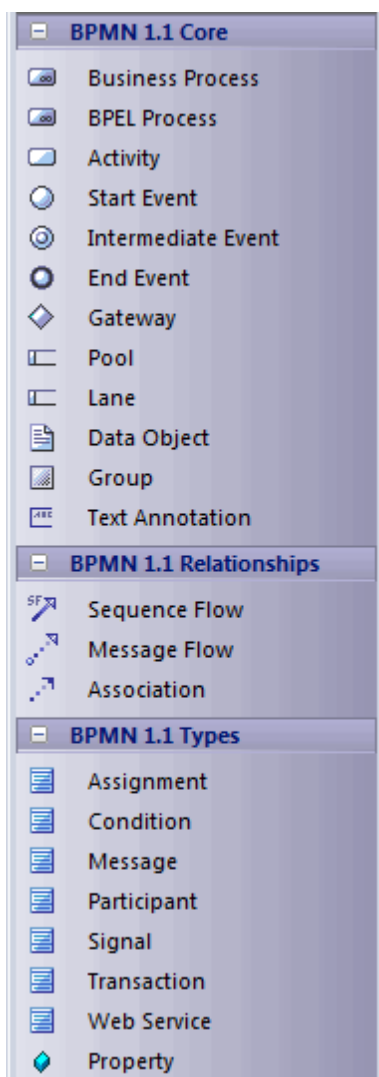
8.3.4.1 BPMN 1.0 and 1.1 Toolbox Pages

The BPMN 1.0 and BPMN 1.1 pages of the Diagram Toolbox enable you to create graphical (Core) and non-graphical (Types) BPMN elements and relationships on business process diagrams in BPMN 1.0, BPMN 1.1 and BPEL formats.

Access [View | Diagram Toolbox: More tools | BPMN 1.0](#)
[View | Diagram Toolbox: More tools | BPMN 1.1](#)


The elements and relationships available are shown below, and described in the following table.

Example



Page	Item	Action	See also
Core	Business Process	Extend a <i>composite Activity</i> that defines a business process.	

Page	Item	Action	See also
	BPEL Process	Define the behavior of an executable or abstract business process.	
	Activity	<p>Represents work that is performed within a <i>Business Process</i>. An <i>Activity</i> can be modeled as a:</p> <ul style="list-style-type: none"> • Sub-Process - a compound <i>Activity</i> that is defined as a flow of other BPMN 1.1 elements or • Task - an atomic <i>Activity</i> that cannot be broken down into a smaller unit <p>As a sub-process, the <i>Activity</i> can be made a composite element that links to a child diagram containing the flow of other BPMN elements.</p>	Composite Elements ^[825]
	Start Event	Define the initiating event in a process.	
	Intermediate Event	Define an intermediate event in a process.	
	End Event	Define the terminating event in a process.	
	Gateway	<p>Define a decision point in a business process.</p> <p>If a condition is true, then processing continues one way; if not, then another.</p>	
	Pool	Extend a <i>Partition</i> element to logically organize an <i>Activity</i> .	
	Lane	Extend a <i>Partition</i> element to subdivide a <i>Pool</i> .	
	Data Object	Extend an <i>Artifact</i> element to define a physical piece of information used or produced by a system.	
	Group	Extend a <i>Boundary</i> element to group other elements.	
	Text Annotation	Create a comment.	


Page	Item	Action	See also
Relationships	Sequence Flow	Extend a <i>Control Flow</i> relationship to define the flow of activity.	
	Message Flow	Extend a <i>Control Flow</i> relationship to define the flow of communications in the process.	
	Association	Associate information and artifacts with flow objects.	
Types	Assignment	Define the <i>properties</i> (Tagged Values) of the Core BPMN 1.1 elements such as Activities, Events and Gates.	Change BPMN Element Appearance 
	Condition		
	Message		
	Participant		
	Signal		
	Rule		
	Transaction		
	Web Service		
	Property		

8.3.4.2 BPMN 2.0 Toolbox Pages

The BPMN 2.0 pages of the Diagram Toolbox enable you to create BPMN elements and relationships on business process diagrams in BPMN 2.0 and BPEL formats.

Access [View | Diagram Toolbox: More tools | BPMN 2.0](#)

Enterprise Architect provides a set of pages for each of the following BPMN 2.0 diagram types:

Topic	Link
Business Process - contains the constructs needed to build Business Process models.	Business Process 

Topic	Link
Choreography - contains the constructs needed to build Choreography models.	Choreography <small>[1553]</small>
Collaboration - contains the constructs required to build Collaboration models.	Collaboration <small>[1553]</small>
Conversation - contains the constructs needed to build Conversation models.	Conversation <small>[1557]</small>
Types - contains the constructs common to all BPMN 2.0 diagrams.	Type <small>[1560]</small>

8.3.4.2.1 BPMN 2.0 Business Process Toolbox Pages

The BPMN 2.0 Business Process pages of the Diagram Toolbox enable you to create BPMN elements and relationships on business process diagrams in BPMN 2.0 and BPEL formats.

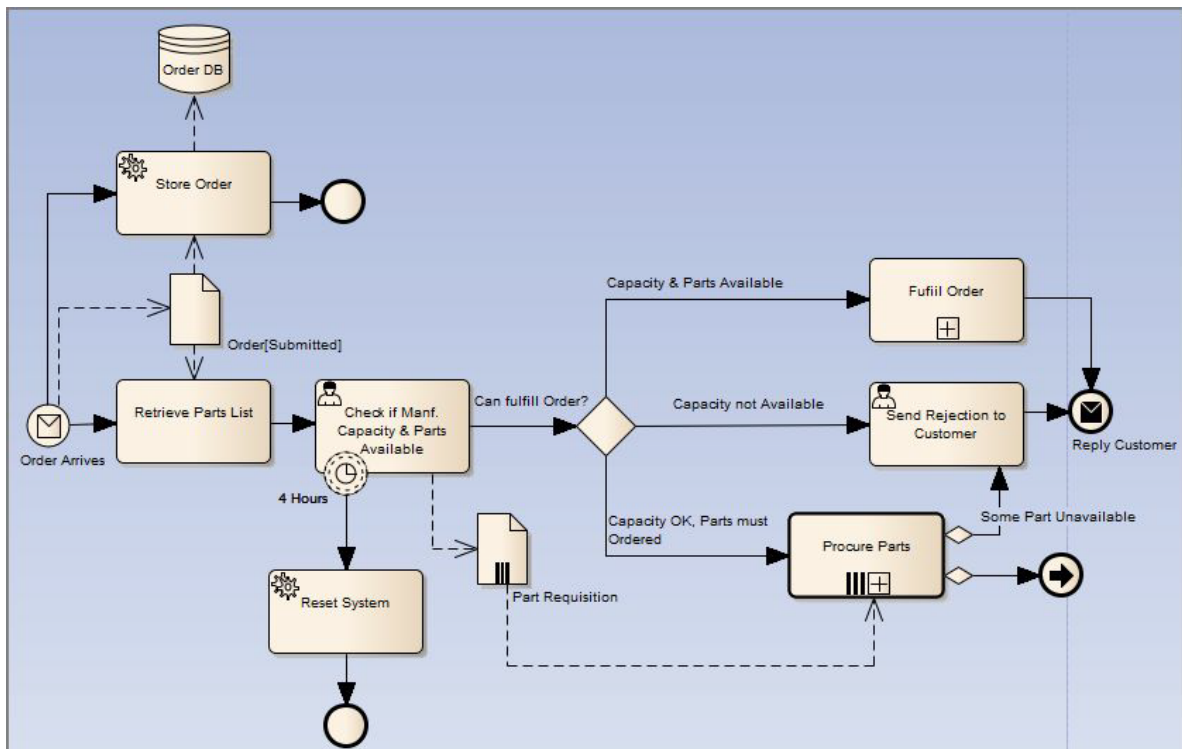
Access [View | Diagram Toolbox: More tools | BPMN 2.0 | BPMN 2.0 Business Process](#)

Page	Item	Use to	See also
BPMN 2.0 Business Process	Business Process	Extend a composite Activity that defines a business process.	
	BPEL	Define the behavior of an executable or abstract business process.	
	Activity	<p>Represents work that is performed within a <i>Business Process</i>. An <i>Activity</i> can be modeled as a:</p> <ul style="list-style-type: none"> • Sub-Process - a compound <i>Activity</i> that is defined as a flow of other BPMN 2.0 elements or • Task - an atomic <i>Activity</i> that cannot be broken down into a smaller unit <p>As a sub-process, the Activity can be made a composite element that links to a child diagram containing the flow of other BPMN elements.</p>	Composite Elements <small>[825]</small>
	Choreography	<p>Extend an Activity element to represent a process unit of information exchange between Participants.</p> <p>Enterprise Architect currently supports two to</p>	

Page	Item	Use to	See also
		four participant bands within one Choreography Activity/Task.	
	Conversation	Extend a Class element to group a set of Message Flows together based on a certain concept.	
	Data Object	Provide or store the information for an Activity.	
	Data Store	Represent a mechanism for an Activity to retrieve or update stored information.	
	Start Event	Define the initiating event in a process.	
	Intermediate Event	Define an intermediate event in a process.	
	End Event	Define the terminating event in a process.	
	Gateway	Define a decision point in a business process. If a condition is true then processing continues one way; if false , then another.	
	Pool	Extend a Partition element to logically organize an Activity.	
	Lane	Extend a Partition element to subdivide a Pool.	
	Message	Represent the contents of a communication between two Participants.	
	Group	Extend a Boundary element to group other elements.	
	Text Annotation	Create a comment.	
BPMN 2.0 Business Process Connectors	Sequence Flow	Extend a Control Flow relationship to define the flow of activity.	

Page	Item	Use to	See also
	Message Flow	Extend a Control Flow relationship to define the flow of communications in the process.	
	Data Association	Move data between Data Objects, Data Store, Properties and Activities, Processes.	
	Association	Link the information and Artifacts with BPMN graphic elements.	
	Conversation Link	Connect a Conversation Node with a Participant, in either direction.	

Example of a BPMN 2.0 Business Process Diagram:



Notes

- The appearance and specification of some elements and connectors are defined by Tagged Values

Learn more

- [Change BPMN Element Appearance](#)^[1561]

Learning Center topics

- (Alt + F1) | [Modeling Languages | Business | BPMN | Business Process Toolbox](#)
- (Alt + F1) | [Business Modeling | BPMN | Process Modeling](#)

8.3.4.2.2 BPMN 2.0 Choreography Toolbox Pages

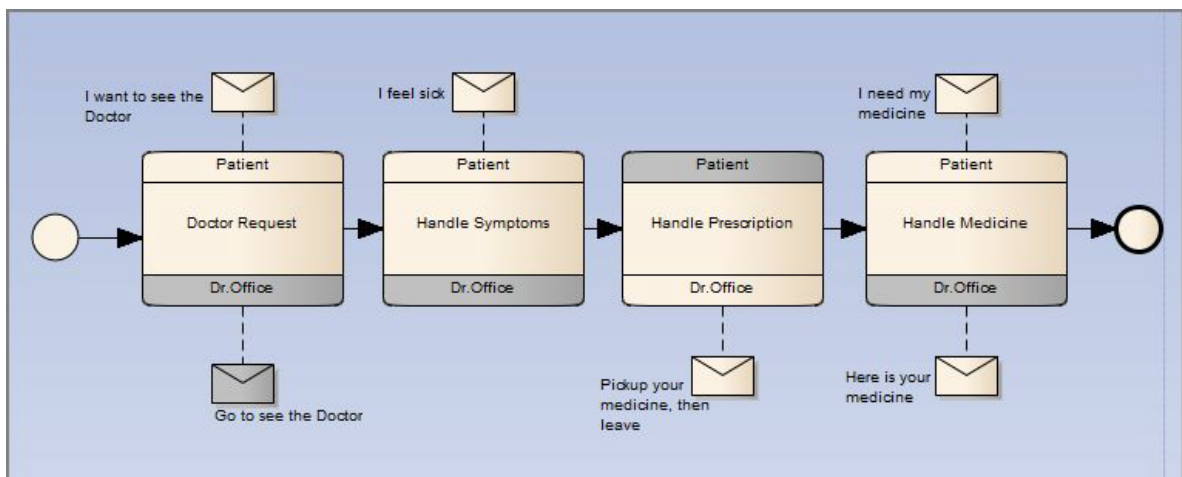
The BPMN 2.0 Choreography pages of the Diagram Toolbox enable you to create BPMN elements and relationships on Choreography diagrams in BPMN 2.0 and BPEL formats.

Access [View | Diagram Toolbox: More tools | BPMN 2.0 | BPMN 2.0 Choreography](#)

Page	Item	Use to
BPMN 2.0 Choreography	Choreography Model	Extend a composite Activity that defines a Choreography process.
	Choreography	Extend an Activity element to represent a process unit of information exchange between Participants. Enterprise Architect currently supports two to four participant bands within one Choreography Activity/Task.
	Start Event	Define the initiating event in a process.
	Intermediate Event	Define an intermediate event in a process.
	End Event	Define the terminating event in a process.
	Gateway	Define a decision point in a business process. If a condition is true then processing continues one way; if false , then another.
	Message	Represent the contents of a communication between two Participants.
	Pool	Extend a Partition element to logically organize an Activity.

Page	Item	Use to
	Text Annotation	Create a comment.
BPMN 2.0 Choreography Connectors	Sequence Flow	Define the order of activity in a Choreography.
	Association	Link the information and Artifacts with BPMN graphic elements.
	Message Flow	Extend a Control Flow relationship to define the flow of communications in the process.

Example of a BPMN 2.0 Choreography Diagram:



Notes

- The appearance and specification of some elements and connectors are defined by Tagged Values

Learn more

- [Change BPMN Element Appearance](#) [156]

Learning Center topics

- **(Alt + F1) | Modeling Languages | Business | BPMN | Choreography**
- **(Alt + F1) | Business Modeling | BPMN | Choreography Modeling**

8.3.4.2.3 BPMN 2.0 Collaboration Toolbox Page

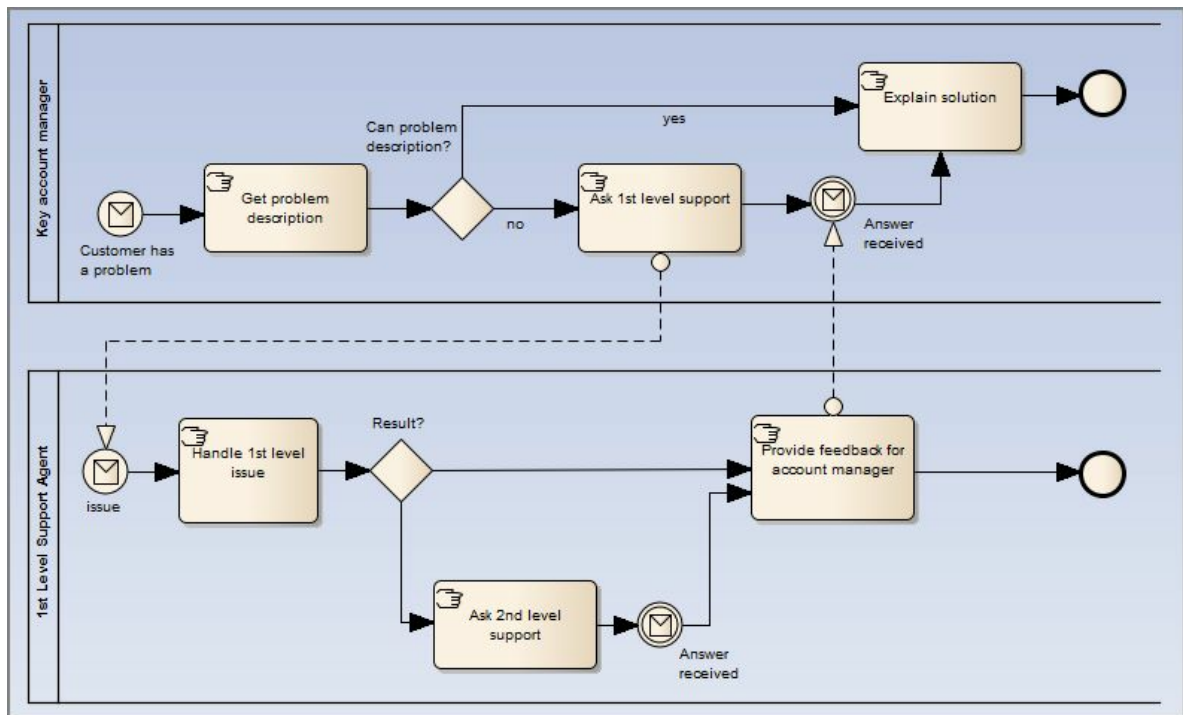
The BPMN 2.0 Collaboration pages of the Diagram Toolbox enable you to create BPMN elements and relationships on Collaboration diagrams in BPMN 2.0 and BPEL formats.

Access [View](#) | [Diagram Toolbox: More tools](#) | [BPMN 2.0](#) | [BPMN 2.0 Collaboration](#)

Page	Item	Use to	See also
BPMN 2.0 Collaboration	Pool	Extend a Partition element to logically organize an Activity.	
	Lane	Extend a Partition element to subdivide a Pool.	
	Conversation	Extend a Class element to group a set of Message Flows based on a certain concept.	
	Start Event	Define the initiating event in a process.	
	Intermediate Event	Define an intermediate event in a process.	
	End Event	Define the terminating event in a process.	
	Gateway	Define a decision point in a business process. If a condition is true then processing continues one way; if false , then another.	
	Activity	Represents work that is performed within a <i>Business Process</i> . An <i>Activity</i> can be modeled as a: <ul style="list-style-type: none"> • Sub-Process - a compound <i>Activity</i> that is defined as a flow of other BPMN 2.0 elements or • Task - an atomic <i>Activity</i> that cannot be broken down into a smaller unit <p>As a sub-process, the Activity can be made a composite element that links to a child diagram containing the flow of other BPMN elements.</p>	Composite Elements ^{B251}
Choreography	Extend an Activity element to represent a process unit of information exchange between participants. Enterprise Architect currently supports two to four		

Page	Item	Use to	See also
		participant bands within one Choreography Activity/ Task.	
	Data Object	Provide or store the information for an Activity.	
	Data Store	Represent a mechanism for an Activity to retrieve or update stored information.	
	Group	Extend a Boundary element to group other elements.	
	Text Annotation	Create a comment.	
BPMN 2.0 Collaboration Connectors	Message Flow	Extend a Control Flow relationship to define the flow of communications in the process.	
	Conversation Link	Connect a Conversation Node to or from a Participant.	
	Sequence Flow	Extend a Control Flow relationship to define the flow of activity.	
	Association	Link the information and artifacts with BPMN graphic elements.	

Example of a BPMN 2.0 Collaboration Diagram:



Notes

- The appearance and specification of some elements and connectors are defined by Tagged Values

Learn more

- [Change BPMN Element Appearance](#)^[156]

Learning Center topics

- (Alt + F1) | [Modeling Languages | Business | BPMN | Collaboration](#)
- (Alt + F1) | [Business Modeling | BPMN | Collaboration Modeling](#)

8.3.4.2.4 BPMN 2.0 Conversation Toolbox Pages

The BPMN 2.0 Conversation pages of the Diagram Toolbox enable you to create BPMN elements and relationships on Conversation diagrams in BPMN 2.0 and BPEL formats.

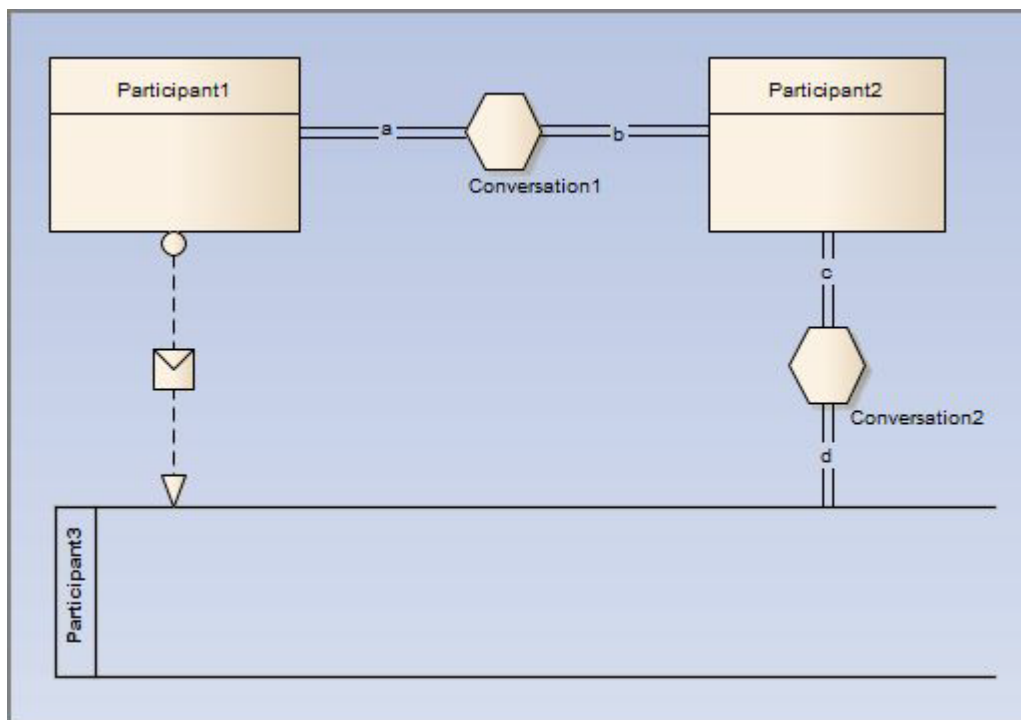
Access [View | Diagram Toolbox: More tools | BPMN 2.0 | BPMN 2.0 Conversation](#)

Page	Item	Use to	See also
BPMN 2.0 Conversation	Conversation Model	Extend a composite Activity that defines the logical relationship of Message exchanges between two or more business entities.	

Page	Item	Use to	See also
	Conversation	Extend a Class element to group a set of Message Flows together based on a certain concept.	
	Participant	Represent a specific organization or individual that is involved in a business process.	
	Pool	Extend a Partition element to logically organize an Activity.	
	Activity	<p>Represents work that is performed within a <i>Business Process</i>. An <i>Activity</i> can be modeled as a:</p> <ul style="list-style-type: none"> • Sub-Process - a compound <i>Activity</i> that is defined as a flow of other BPMN 2.0 elements or • Task - an atomic <i>Activity</i> that cannot be broken down into a smaller unit <p>As a sub-process, the Activity can be made a composite element that links to a child diagram containing the flow of other BPMN elements.</p>	Composite Elements ⁸²⁵
	Start Event	Define the initiating event in a process.	
	Intermediate Event	Define an intermediate event in a process.	
	End Event	Define the terminating event in a process.	
	Gateway	Define a decision point in a business process. If a condition is true then processing continues one way; if false , then another.	
	Text Annotation	Create a comment.	
BPMN 2.0 Conversation Connectors	Conversation Link	Connect a Conversation Node to or from a Participant.	
	Message Flow	Extend a Control Flow relationship to define the	

Page	Item	Use to	See also
		flow of communications in the process.	
	Sequence Flow	Extend a Control Flow relationship to define the flow of activity.	
	Association	Link the information and artifacts with BPMN graphic elements.	

Example of a BPMN 2.0 Conversation Diagram:



Notes

- The appearance and specification of some elements and connectors are defined by Tagged Values

Learn more

- [Change BPMN Element Appearance](#)^[156]


Learning Center topics

- (Alt + F1) | [Modeling Languages](#) | [Business](#) | [BPMN](#) | [Conversation](#)
- (Alt + F1) | [Business Modeling](#) | [BPMN](#) | [Conversation Modeling](#)

8.3.4.2.5 BPMN 2.0 Type Toolbox Page

The BPMN 2.0 Types page of the Diagram Toolbox is common to all BPMN 2.0 diagrams.

Access [View](#) | [Diagram Toolbox: More tools](#) | [BPMN 2.0](#) | [BPMN 2.0 <diagram type>](#)

Page	Item	Use To
BPMN 2.0 Types	Assignment	Define the non-graphic elements or properties (Tagged Values) of the Core BPMN 2.0 graphic elements. See Change BPMN Element Appearance  .
	Correlation Key	
	Correlation Property	
	Correlation Property Binding	
	Correlation Property Retrieval	
	Correlation Subscription	
	Complex Behavior Definition	
	Item Definition	
	Error	
	Escalation	
	Signal	
	IO Specification	
	Input Set	
	Output Set	
	Input Output Binding	
	Interface	
	Operation	
	Participant	
	Resource	
Resource Role		
Resource Parameter		
Property		

Page	Item	Use To
	Participant Association	
	Conversation Association	

8.3.4.3 Change BPMN Element Appearance

A number of Tagged Values directly affect the appearance of the BPMN element or relationship they apply to. Therefore, by changing the value of the tag you can change the appearance of the object.

Access [View | Tagged Values](#)

How to

To define the appearance of BPMN elements and relationships

Step	Action	See also
1	Open the Tagged Values window.	Tagged Values [969]
2	Select the required element or relationship in a diagram. The Tagged Values window shows the appropriate Tagged Values and provides a list of values to assign to each one.	
3	Set the required value, based on the value descriptions in the table below.	

Reference

Element	Tags	Values Setting
Activity	<i>activityType</i>	Set to: <ul style="list-style-type: none"> • Task for no decoration • Sub-Process to display the <i>plus-in-box</i> decoration on the bottom edge of the shape
	<i>adHoc</i>	Set to true to display the <i>tilde</i> decoration on the bottom edge of the shape, indicating that this is an ad-hoc Activity.
	<i>adHocOrdering</i>	Set to: <ul style="list-style-type: none"> • Parallel to display three vertical lines on the bottom edge of the shape • Sequential to display three horizontal lines on the

Element	Tags	Values Setting
		<p>bottom edge of the shape</p> <ul style="list-style-type: none"> • Unspecified to display no lines on the bottom edge of the shape. <p>The <i>adHoc</i> tag must first be set to true.</p>
	<i>isACallActivity</i>	Set to true to show the boundary of the Activity element as a thick line, indicating that this is a Call Activity.
	<i>isATransaction</i>	Set to true to give the Activity element a double-lined border, indicating that this is a Transaction.
	<i>isForCompensation</i>	Set to true to display the <i>rewind</i> icon on the bottom edge of the shape, indicating that this is a Compensation Activity.
	<i>isSequential</i>	<p>Set to true to display three horizontal bars on the bottom edge of the shape.</p> <p>The <i>loopCharacteristics</i> tag must first be set to Multilinstance.</p>
	<i>loopCharacteristics</i>	<p>Set to:</p> <ul style="list-style-type: none"> • Standard to display three vertical bars at the bottom edge of the shape • Multilinstance to display a loop icon at the bottom edge of the shape
	<i>taskType</i>	When set to different value, the associated decoration is displayed at the top-left corner of the shape.
	<i>triggeredByEvent</i>	Set to true to show the boundary of the Activity element as a dotted line.
Choreography	<i>choreographyType</i>	<p>Set to:</p> <ul style="list-style-type: none"> • Task for no decoration • Sub-Process to display the <i>plus-in-box</i> decoration on the bottom edge of the shape.
	<i>initiatingParticipant</i>	Set a value to show the corresponding participant band in a light color, which indicates that this is the initiating participant.

Element	Tags	Values Setting
	<i>isACallChoreography</i>	Set to true show the boundary of the Choreography element as a thick line.
	<i>loopType</i>	<p>Set to:</p> <ul style="list-style-type: none"> • Standard to display three vertical bars at the bottom edge of the shape • Multinstance to display a loop icon at the bottom edge of the shape <p>Each Participant has its own MultInstance setting.</p>
	<i>numberOfParticipants</i>	Set to the appropriate numerical value to indicate the number of Participant bands for a choreography Activity or Task.
	<i>participantA</i> or <i>participantARef</i>	<p><i>participantA</i> - Type the name of the Participant to show on the Participant band of the Choreography.</p> <p>If the Participant is known and referenced, use the <i>participantARef</i> tag and leave the <i>participantA</i> tag blank; the name of the referenced Participant is shown on the Participant band of the Choreography.</p> <p>The same usage is applied to other Participant bands.</p>
Conversation	<i>isACallConversation</i>	Set to true to show the boundary of the Conversation element as a thick line.
	<i>isComposite</i>	Set to true to display a <i>plus-in-box</i> decoration on the bottom edge of the shape, indicating that the Conversation is a composite element.
Data Object	<i>dataInOut</i>	<p>Set to:</p> <ul style="list-style-type: none"> • Input to add a light-colored arrow at the top-left corner of the Data Object, indicating that it represents a Data Input element • Output to add a dark-colored arrow at the top-left corner of the Data Object, indicating that it represents a Data Output element
	<i>isCollection</i>	Set to true to add three vertical lines at the bottom edge of the shape, indicating that the object is a collection.

Element	Tags	Values Setting
Event	<i>cancelActivity</i>	For Intermediate Event only. Set to false to show the Intermediate Event border as a dashed line.
	<i>catchOrThrow</i>	For Intermediate Event only. Set to: <ul style="list-style-type: none"> • Catch or • Throw
	<i>eventDefinition</i>	Use to change the decoration of a Start Event, Intermediate Event or End Event.
	<i>isInterrupting</i>	For Start Event only. Set to false to show the border of the Start Event as a dashed line, indicating that the Start Event is not an interrupting event.
Gateway	<i>eventGatewayType</i>	Use to define two types of Event-based Gateway: <ul style="list-style-type: none"> • Exclusive and • Parallel
	<i>gatewayType</i>	Set to: <ul style="list-style-type: none"> • Exclusive to render the Gateway as a diamond shape without any marker in the middle; if you also set the <i>markerVisible</i> tag to true, the Gateway is rendered as a diamond shape with an X inside • Complex to render the Gateway as a diamond shape with an asterisk (*) inside • Inclusive to render the Gateway as a diamond shape with a circle (o) inside • Parallel to render the Gateway as a diamond shape with a plus (+) inside • Event to define the Gateway as Event-based.
	<i>instantiate</i>	Use only to define the Exclusive and Parallel Event-based Gateway.
Message	<i>IsInitiate</i>	Set to true to make the envelope color gray, indicating that this is an initiating Message.
Message Flow	<i>messageVisible</i>	Set to:

Element	Tags	Values Setting
		<ul style="list-style-type: none"> • Unspecified to make the Message Flow connector without decoration • Initiating to add a white envelope to the Message Flow connector • Non-Initiating to add a gray envelope to the Message Flow connector
Participant	<i>participantMultiplicity</i>	<p>Set to true to display three vertical bars at the bottom edge of a rectangle Pool element.</p> <p>Participant is rendered as a normal UML Class shape exclusively on Conversation diagrams.</p>
Pool	<i>blackBoxPool</i>	Set to true to render the Pool element as a rectangle, which is used in a Main Pool diagram.
	<i>participantMultiplicity</i>	Set to true to display three vertical bars at the bottom edge of a rectangle Pool element.
Sequence Flow	<i>conditionType</i>	<p>Set to:</p> <ul style="list-style-type: none"> • Expression to show an unfilled diamond marker at the source end of the connector • Default to show a slash marker at the source end of the connector

Notes

- Some BPMN elements changed in appearance between BPMN version 1.0 and BPMN version 1.1
- In Enterprise Architect releases later than 7.1, if you work on a BPMN 1.0 model created in an earlier release, existing elements default to their version 1.0 appearance; new elements assume the BPMN version 1.1 appearance and automatically have a Tagged Value BPMNVersion set to 1.1
- If you want a new element to revert to the BPMN version 1.0 appearance, set the BPMNVersion Tagged Value to 1.0
- Conversely, if you want an older element to assume the BPMN version 1.1 appearance, assign the BPMNVersion Tagged Value to it, with the value 1.1

8.3.4.4 Migrate BPMN 1.0 Model to BPMN 1.1

Enterprise Architect enables you to migrate a BPMN 1.0 model (or part of a model) to BPMN 1.1 or BPMN1.1::BPEL, using the Automation Interface. There are two functions you can use:

- *MigrateToBPMN11()*

- `proj.MigrateToBPMN11 sGUID, " BPMN"`
- `proj.MigrateToBPMN11 sGUID, " BPEL"`
- *Migrate()*
 - `proj.Migrate sGUID, " BPMN", " BPMN1.1"`
 - `proj.Migrate sGUID, " BPMN", " BPMN1.1:: BPEL"`

These functions update the Tagged Values and, if required, stereotypes to BPMN 1.1 for all elements, attributes, connectors and diagrams under the selected package or element.

Example

The following VB script calls the *MigrateToBPMN11()* function to migrate the Tagged Values to BPMN 1.1:

```

Sub MigrateElement (sGUID, lngPackageID)

    Dim proj as EA.Project
    set proj = Repository.GetProjectInterface
    proj.MigrateToBPMN11 sGUID, " BPMN"

    ' refresh the model
    If lngPackageID <> 0 Then
        Repository.RefreshModelView (lngPackageID)
    End If

End Sub

Sub MigrateSelectedItem

    Dim selType
    Dim selElement as EA.Element
    Dim selPackage as EA.Package

    selType = GetTreeSelectedItemType

    If selType = 4 Then ' means Element
        set selElement = GetTreeSelectedObject
        MigrateElement selElement.ElementGUID, selElement.PackageID
        MsgBox " Complete", 0, " BPMN 1.1 Migration"

    ElseIf selType = 5 Then ' means Package
        set selPackage = GetTreeSelectedObject
        MigrateElement selPackage.PackageGUID, selPackage.PackageID
        MsgBox " Complete", 0, " BPMN 1.1 Migration"

    Else
        MsgBox " Select a Package or Element in the Project Browser to
        initiate migration", 0, " BPMN 1.1 Migration"

    End If

End Sub

Sub Main

    MigrateSelectedItem

End Sub

Main

```

Notes

- In BPMN 1.0, various tags have free-text direct-entry value fields, and you can provide additional information on these tags in the Tagged Value Note dialog, for display at the bottom of the Tagged Values window

In BPMN 1.1, some of these tags (such as the *Categories* tag on a *BusinessProcess* stereotyped element) have been changed to memo type, and you use the Tagged Value Note dialog to enter the value; therefore, you cannot have additional notes for these tags, all information must be within the tag's value

- For such tags, when migrating from BPMN 1.0 to BPMN 1.1, the BPMN 1.0 tag value is moved into the BPMN 1.1 tag notes field and the BPMN 1.0 tag notes are discarded; if you want to preserve the tag notes text, take a copy of the BPMN 1.0 model before migration to enable you to copy the tag notes text into the tag value after migration

Learn more

- [MigrateToBPMN11\(\)](#)^[2554]
- [Migrate\(\)](#)^[2553]

8.3.4.5 Migrate BPMN 1.1 Model to BPMN 2.0

Enterprise Architect enables you to migrate a BPMN 1.1 model (or part of a model) to BPMN 2.0, using the Automation Interface function *Migrate()*. This function updates the Tagged Values and, if required, stereotypes to BPMN 2.0 for all elements, attributes, connectors and diagrams under the selected package or element.

Example

The following VB script calls the *Migrate()* function to migrate the package or element to BPMN 2.0:

```
Sub MigrateElement (sGUID, lngPackageID)
    Dim proj as EA.Project
    set proj = Repository.GetProjectInterface
    proj.Migrate sGUID, "BPMN1.1", "BPMN2.0"

    'refresh the model
    If lngPackageID > 0 Then
        Repository.RefreshModelView (lngPackageID)
    End If
End Sub

Sub MigrateSelectedItem
    Dim selType
    Dim selElement as EA.Element
    Dim selPackage as EA.Package
    selType = GetTreeSelectedItemType
    If selType = 4 Then 'means Element
        set selElement = GetTreeSelectedObject
        MigrateElement selElement.ElementGUID, selElement.PackageID
        MsgBox "Element Migration Completed", 0, "BPMN 2.0 Migration"
    ElseIf selType = 5 Then 'means Package
        set selPackage = GetTreeSelectedObject
    End If
End Sub
```

```

        MigrateElement sel Package.PackageGUID, sel Package.PackageID
        MsgBox "Package Migration Completed", 0, "BPMN 2.0 Migration"
    Else
        MsgBox "Select a Package or Element in the Project Browser to
        initiate migration", 0, "BPMN 2.0 Migration"
    End If
End Sub

Sub Main
    MigrateSelectedItem
End Sub

Main

```

Notes

- Please backup your project before you run the BPMN 2.0 Migrator
- Normalization occurs on the following tags:
 - BPMN1.1::Activity::InMessageRef
 - BPMN1.1::Activity::OutMessageRef
 - BPMN1.1::Activity::IORules
 - BPMN1.1::Activity::InputSets
 - BPMN1.1::Activity::OutputSets
 - BPMN1.1::Activity::ComplexMI_FlowCondition
 - BPMN1.1::Activity::Performers
 - BPMN1.1::BusinessProcess::InputSets
 - BPMN1.1::BusinessProcess::OutputSets
 - BPMN1.1::BusinessProcess::Performers
 - BPMN1.1::EndEvent::ErrorCode
 - BPMN1.1::IntermediateEvent::ErrorCode

Taking *BPMN1.1::Activity::InMessageRef* as an example, the following steps take place:

- (1) Create a new element *BPMN2.0::Operation*
- (2) Insert the reference tag *BPMN2.0::Activity::operationRef*
- (3) Migrate *BPMN1.1::Activity::InMessageRef* to *BPMN2.0::Operation::InMessageRef*

- Denormalization occurs on the following tags:
 - BPMN1.1::Transaction::TransactionProtocol
 - BPMN1.1::WebService::Operation
 - BPMN1.1::WebService::Interface
 - BPMN1.1::WebService::ParticipantRef
 - BPMN1.1::Condition::ConditionExpression
 - BPMN1.1::BPELProcess::InputSets
 - BPMN1.1::BPELProcess::OutputSets

Taking *BPMN1.1::Transaction::TransactionMethod* as an example, the following steps take place:

- (1) Find element *BPMN1.1::Transaction* by '*BPMN1.1::Transaction::TransactionProtocol*
- (2) Migrate *BPMN1.1::Transaction::TransactionMethod* to *BPMN2.0::transactionMethod*
- (3) Migrate *BPMN1.1::Transaction::TransactionProtocol* to *BPMN2.0::transactionProtocol*

- If a BPMN1.1 stereo tag is discarded in BPMN2.0, it is preserved if the tag has a value

Learn more

- [Migrate\(\)](#)^[2553]

8.3.4.6 BPMN 2.0 XML

Enterprise Architect supports serializing **BPMN 2.0 Models** in **BPMN 2.0 XML**. The serialized XML file contains both the model semantics as well as the diagram-interchange information.

Access Click on a package in the Project Browser, then:

Project | Model Publisher
Project | Model Import/Export | Export Package to XMI File: Publish, or
Right-click | Import/Export | Export Package to XMI: Publish

How to

Step	Action	See also
1	Open the Publish Model Package dialog.	Publish Model Package [414]
2	In the XML Type field, select BPMN 2.0 XML .	
3	Click on the Export button to initiate the BPMN 2.0 XML serialization.	

Notes

- The package being serialized is assumed to be self-contained
- Only the contents of the selected package are considered for serialization - child packages (and their contents) are ignored
- Serialization of the Group element in the BPMN 2.0 Business Process Tool-box is not supported
- Serialization of the following items in the BPMN 2.0 Types section of the Tool-box is not supported:
 - *InputOutputBinding*
 - *Participant Association*
 - *Conversation Association*

Learn more

- [XML Schema for BPMN 2.0 XML](#) (Online Resource)
- [BPMN 2.0 Toolbox Pages](#)^[1549]

8.3.5 BPEL Models

The following text is derived from the **BPEL** entry in the online Wikipedia:

Business Process Execution Language (BPEL), short for **Web Services Business Process Execution Language (WS-BPEL)**, is an executable language for specifying interactions with **Web Services**. Processes in Business Process Execution Language export and import information by using Web Service interfaces exclusively.

Web service interactions can be described in two ways :

1. Executable business processes, which model the actual behavior of a participant in a business interaction.
2. Abstract business processes, which are partially specified processes that are not intended to be executed. An Abstract Process may hide some of the required concrete operational details.

BPEL is an **Orchestration** language, serialized in XML, which specifies an executable process that involves message exchanges with other systems. This messaging facility depends on the use of the **Web Services Description Language (WSDL) 1.1** to describe outgoing and incoming messages.

BPEL in Enterprise Architect:

Enterprise Architect supports generating **BPEL** from executable processes in the **Business and Software Engineering** and the **Ultimate** editions. Although there is no standard graphical notation for **WS-BPEL**, Enterprise Architect uses **BPMN** profile as a graphical front-end to capture **BPEL** Process descriptions. With the help of the **BPMN** Profile, Enterprise Architect enables you to develop **BPEL** diagrams quickly and simply. While **BPMN** provides a graphical notation for visualizing business process, **BPEL** provides a way for visualizing this graphical business process in XML.

Enterprise Architect supports generating:

- BPEL 1.1^[1571] from models created using constructs in BPMN 1.1 Profile
- BPEL 2.0^[1595] from models created using constructs in BPMN 2.0 Profile

Notes

- BPEL is supported in the **Business and Software Engineering** and the **Ultimate** editions of Enterprise Architect
- Enable BPMN 1.1 Technology for BPEL 1.1 and BPMN 2.0 Technology for BPEL 2.0 modeling in the MDG Technologies dialog (**Settings | MDG Technologies**)

Learn more

- [BPEL](#) (Online Resource)
- [Orchestration](#) (Online Resource)

- [BPEL 1.1 Model](#)^[1571]
- [BPEL 2.0 Model](#)^[1595]
- [BPMN Model](#)^[1545]

Learning Center topics

- **(Alt + F1) | Modeling Languages | Business | BPMN | BPEL**
- **(Alt + F1) | Business Modeling | BPEL | Getting Started**

8.3.5.1 BPEL 1.1 Model

Enterprise Architect uses **BPMN 1.1** as a graphical front-end to capture **BPEL 1.1** Process description (as BPEL 1.1 does not have not a standard graphical notation). Enterprise Architect uses the partial mapping specified in the *OMG BPMN 1.1 Specification* to map BPMN 1.1 constructs to BPEL 1.1.

How to

Step	Action	See also										
1	<p>Create a BPEL 1.1 Package Structure.</p> <p>Alternative:</p> <table border="1"> <thead> <tr> <th>Step</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>a</td> <td>Create a package in the Project Browser.</td> </tr> <tr> <td>b</td> <td>Click on the New Diagram icon in the Project Browser.</td> </tr> <tr> <td>c</td> <td>Select BPMN 1.1 in the Select From field and BPEL in the Diagram</td> </tr> <tr> <td>d</td> <td>Click on the BPEL Process icon from the BPMN 1.1 Core toolbox and the created BPEL diagram.</td> </tr> </tbody> </table>	Step	Action	a	Create a package in the Project Browser.	b	Click on the New Diagram icon in the Project Browser.	c	Select BPMN 1.1 in the Select From field and BPEL in the Diagram	d	Click on the BPEL Process icon from the BPMN 1.1 Core toolbox and the created BPEL diagram.	<p>BPEL 1.1 Package Structure^[1573]</p> <p>Project Browser Toolbar^[583]</p>
Step	Action											
a	Create a package in the Project Browser.											
b	Click on the New Diagram icon in the Project Browser.											
c	Select BPMN 1.1 in the Select From field and BPEL in the Diagram											
d	Click on the BPEL Process icon from the BPMN 1.1 Core toolbox and the created BPEL diagram.											
2	Open the BPEL diagram under the <i>BPELProcess</i> stereotyped element.											
3	<p>Model the <i>BPEL Process</i> using the constructs in the BPMN 1.1 Core toolbox; use the following constructs in this toolbox to model BPEL 1.1:</p> <p>BPMN 1.1 - Core page:</p> <ul style="list-style-type: none"> • BPEL Process • Activity • Start Event 	<p>BPMN 1.1 Core Toolbox^[1547]</p> <p>Model BPEL 1.1 Process^[1576]</p>										

	<ul style="list-style-type: none"> • Intermediate Event • End Event • Gateway • Pool <p>BPMN 1.1 - Relationships page:</p> <ul style="list-style-type: none"> • Sequence Flow • Message Flow • Association <p>BPMN 1.1 Types page:</p> <ul style="list-style-type: none"> • Assignment • Condition • Message • Participant • Web Service 	
4	Create Web Service for the <i>BPEL Process</i> and other related <i>Pools</i> , if required.	BPEL Web Service <small>[1591]</small>
5	Generate BPEL 1.1 code from the <i>BPEL Process</i> .	Generate BPEL 1.1 <small>[1594]</small>

Notes

- Generating BPEL 1.1 from a BPMN 1.1 model is supported in the Business and Software Engineering and the Ultimate editions of Enterprise Architect
- Ensure that BPMN 1.1 Technology is enabled in the MDG Technologies dialog (**Settings | MDG Technologies**)

BPEL 1.1 Example:

- The Enterprise Architect Example file (*EAExample.EAP*) has a sample BPMN 1.1 model from which BPEL 1.1 can be generated
- If you have installed Enterprise Architect at the default location, open this file: *C:\Program Files\Sparx Systems\EA\EAExample.EAP*
- The BPMN 1.1 model package is in: **Project Models | Analysis and Business Modeling | BPEL Example | BPEL 1.1 Model**

Modeling Restrictions:

- Every *BPEL Process* and *Sub-Process* should start with a *Start Event* and end with an *End Event*
 - A *Start Event* or an *End Event* should not be attached to the boundary of a *Sub-Process*
- *Sequence Flow* Looping is not supported - only *Activity* looping is supported; all *Sequence Flows* should flow downstream and not upstream
- Mapping of an *Intermediate Event* with multiple triggers to BPEL 1.1 is not supported
- Mapping of *Multi-Instance Parallel While* loops to BPEL 1.1 is not supported
- Mapping of *Independent Sub-Process* to **BPEL 1.1** is not supported
- *Pools* are treated as *black box* (i.e. they do not expose any details) and hence they cannot contain any child elements or have any incoming/outgoing *Sequence Flow* connectors

8.3.5.1.1 Create BPEL 1.1 Model Structure

A BPEL 1.1 model consists of a *BPEL Process* (containing a BPEL diagram and mappable BPMN 1.1 constructs) and other supporting elements (like BPMN 1.1 *Assignment*, BPMN 1.1 *Web Service*) required for generating BPEL 1.1 code. A sample BPEL 1.1 package structure can be created in the Project Browser, using the Select Model(s) (Model Wizard) dialog. You can use this package structure as a template for developing your *BPEL Process*.

Access

- **Project Browser package context menu | Add | Add a New Model using Wizard** (or)
- **Project Browser toolbar: New Model from Pattern**

Use to

- Create a new **BPEL 1.1** package structure that can be used as a starting point for developing your *BPEL Process*.

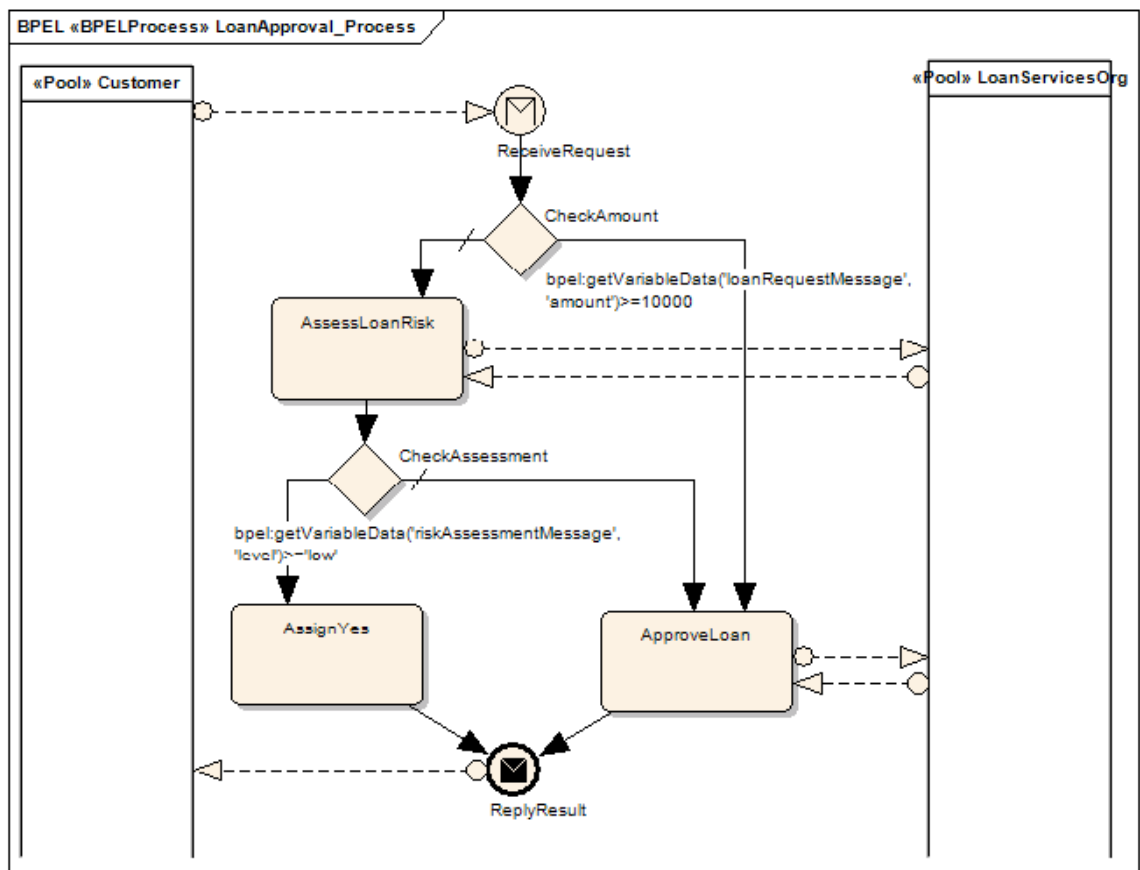
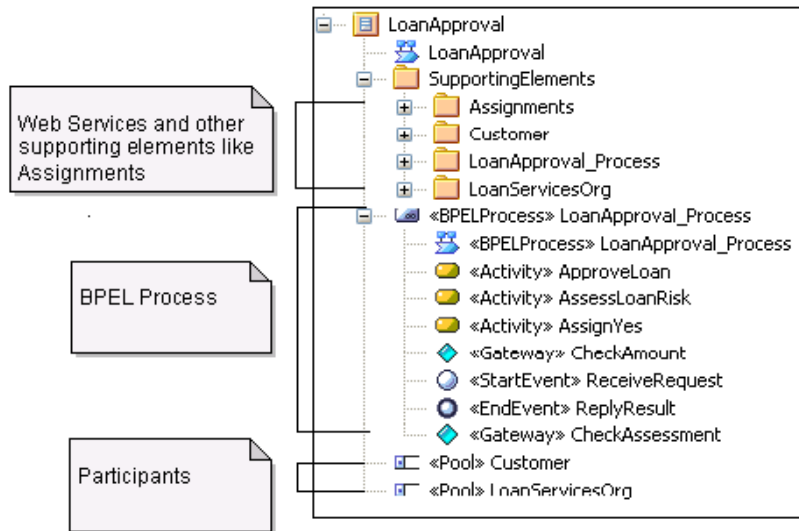
How to

Step	Action	See also
1	Select the root node or a package in the Project Browser.	
2	Click on the New Model from Pattern icon in the toolbar.	Project Browser Toolbar ^[583]
3	Select the value BPMN 1.1 in the Technology section.	Model Wizard ^[660]
4	Check the option BPEL 1.1 Model in the Name section.	
5	Click on the OK button to create the sample BPEL 1.1 package structure.	

--	--	--

Example BPEL 1.1 Package Structure:

The BPEL Process *LoanApproval_Process* acts as container for the BPEL diagram and elements. The *SupportingElements* package contains supporting components like **Assignments** and **Web Services**.



Learn more

- [Model a BPEL 1.1 Process](#) ^[1578]
- [Create a BPEL 1.1 Web Service](#) ^[1591]
- [Generate BPEL 1.1](#) ^[1594]

8.3.5.1.2 Model a BPEL 1.1 Process

The *BPEL Process* in Enterprise Architect represents the top-level container for the BPMN 1.1 elements, from which BPEL 1.1 can be generated. Conceptually it maps to the BPEL *process* element.

BPEL Properties:

- Double-click on the *BPEL Process* in the BPEL diagram
- Right-click on the *BPEL Process* | **BPEL** | **BPEL Properties**

Reference

Field/Button	Usage	See also
Name	Specify the name for the <i>BPEL Process</i> .	
Query Language	Specify the value of the language used in the <i>BPEL Process</i> for the selection of nodes in <i>Assignments</i> . Defaults to XPath 1.0 .	
OK	Save the values entered in the dialog.	
Cancel	Discard the values entered in the dialog.	
Help	Display this Help topic.	
General	Open the UML Properties dialog.	Properties [843]

Learn more

The *BPEL Process* element acts as a container for the following BPMN 1.1 constructs:

- [Start Event](#) [1577]
- [Intermediate Event](#) [1578]
- [Activity](#) [1581]
- [Gateway](#) [1584]
- [End Event](#) [1586]
- [Sequence Flow](#) [1587]

The following BPMN 1.1 constructs can be optionally used in BPEL modeling:

- [Pool](#) [1588]

- [Assignment](#)^[1589]

8.3.5.1.2.1 Start Event

A BPMN 1.1 *Start Event* indicates where a particular *Process* begins. Every *Process* in Enterprise Architect must begin with a *Start Event*.

BPEL Properties:

- Double-click on the *Start Event* in the BPEL diagram (or)
- Right-click on the *Start Event* | **BPEL** | **BPEL Properties**

Reference

Field/Button	Usage	See also										
Name	Specify the name for the <i>Start Event</i> .											
Trigger Type	Select a trigger for the <i>Start Event</i> , namely: <ul style="list-style-type: none"> • Conditional • Link • Message • Multiple • None • Signal • Timer 											
Details	Depending on the selected Trigger Type , the Details tab changes as follows: <table border="1" data-bbox="491 1503 1230 2016"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Conditional Trigger Type</td> </tr> <tr> <td>Condition</td> <td>Select a BPMN 1.1 <i>Condition</i> from the list of Co created in the <i>SupportingElements</i> package.</td> </tr> <tr> <td colspan="2">Message Trigger Type</td> </tr> <tr> <td>Web Service</td> <td>Select a BPMN 1.1 <i>Web Service</i> from the packe (<i>SupportingElements</i> package) that represents involved <i>Participants</i>.</td> </tr> </tbody> </table>	Field	Usage	Conditional Trigger Type		Condition	Select a BPMN 1.1 <i>Condition</i> from the list of Co created in the <i>SupportingElements</i> package.	Message Trigger Type		Web Service	Select a BPMN 1.1 <i>Web Service</i> from the packe (<i>SupportingElements</i> package) that represents involved <i>Participants</i> .	
Field	Usage											
Conditional Trigger Type												
Condition	Select a BPMN 1.1 <i>Condition</i> from the list of Co created in the <i>SupportingElements</i> package.											
Message Trigger Type												
Web Service	Select a BPMN 1.1 <i>Web Service</i> from the packe (<i>SupportingElements</i> package) that represents involved <i>Participants</i> .											

	Message	Select a BPMN 1.1 <i>Message</i> from the list of <i>Me</i> the selected Web Service.	
	Multiple Trigger Type		
	Events	Select additional BPMN 1.1 <i>Start Events</i> from the events in this <i>Process</i> that might trigger the <i>Pro</i> clicking the (...) button.	
	Timer Trigger Type		
	Time Cycle	Specify the value for the time duration.	
	Time Date	Specify the value for the time date.	
	Assignments	Select one or more <i>Assignment</i> elements created in the <i>SupportingElements</i> package in this tab (Optional).	
OK	Save the values entered in the dialog.		
Cancel	Discard the values entered in the dialog.		
Help	Display this Help topic.		
General	Open the UML Properties dialog.	Properties ^[843]	

Notes

- Either set **Time Cycle** or **Time Date** for **Timer Trigger Type** but not both as they are mutually exclusive fields
- **Link**, **None** and **Signal Event** types cannot be mapped to BPEL 1.1

8.3.5.1.2.2 Intermediate Event


A BPMN 1.1 *Intermediate Event* indicates where an event occurs somewhere between the start and end of a *Process*.

BPEL Properties:

- Double-click on the *Intermediate Event* in the BPEL diagram
- Right-click on the *Intermediate Event* | **BPEL** | **BPEL Properties**

Reference

Field/Button	Usage	See also														
Name	Specify the name for the <i>Intermediate Event</i> .															
Trigger Type	Select the type for the <i>Intermediate Event</i> , namely: <ul style="list-style-type: none"> • Cancel • Compensation • Conditional • Error • Link • Message • Multiple • None • Signal • Timer 															
Details	Depending on the selected Trigger Type , the Details tab changes as follows: <table border="1" data-bbox="491 1346 1214 2033"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Compensation Trigger Type</td> </tr> <tr> <td>Activity</td> <td>Select a BPMN 1.1 <i>Task</i> from the list of all Activities in the <i>Process</i>.</td> </tr> <tr> <td colspan="2">Conditional Trigger Type</td> </tr> <tr> <td>Condition</td> <td>Select a BPMN 1.1 <i>Condition</i> from the list of (s) created in the <i>SupportingElements</i> package.</td> </tr> <tr> <td colspan="2">Error Trigger Type</td> </tr> <tr> <td>Error Code</td> <td>Specify the required error code.</td> </tr> </tbody> </table>	Field	Usage	Compensation Trigger Type		Activity	Select a BPMN 1.1 <i>Task</i> from the list of all Activities in the <i>Process</i> .	Conditional Trigger Type		Condition	Select a BPMN 1.1 <i>Condition</i> from the list of (s) created in the <i>SupportingElements</i> package.	Error Trigger Type		Error Code	Specify the required error code.	Create BPEL 1.1 Model Structure <small>[1573]</small>
Field	Usage															
Compensation Trigger Type																
Activity	Select a BPMN 1.1 <i>Task</i> from the list of all Activities in the <i>Process</i> .															
Conditional Trigger Type																
Condition	Select a BPMN 1.1 <i>Condition</i> from the list of (s) created in the <i>SupportingElements</i> package.															
Error Trigger Type																
Error Code	Specify the required error code.															

	Link Trigger Type	
	Link Event	Select a <i>Link Event</i> element (to act as a target <i>Link Event</i>) from the list under the current <i>Process</i> or <i>Sub-Process</i> .
	Message Trigger Type	
	Web Service	Select a BPMN 1.1 <i>Web Service</i> from the package (from the <i>SupportingElements</i> package) that represents the involved <i>Participants</i> .
	Message	Select a BPMN 1.1 <i>Message</i> from the list of Messages in the selected <i>Web Service</i> .
	Multiple Trigger Type	
	Events	Select additional BPMN 1.1 <i>Intermediate Event</i> from the list of events in this <i>Process</i> by clicking the Events button.
	Timer Trigger Type	
	Time Cycle	Specify the value for the time duration.
	Time Date	Specify the value for the time date.
OK	Save the values entered in the dialog.	
Cancel	Discard the values entered in the dialog.	
Help	Display this Help topic.	
General	Open the UML Properties dialog.	Properties 

Notes

- **Cancel**, **None** and **Signal** *Event* types cannot be mapped to BPEL 1.1
- Either set **Time Cycle** or **Time Date** for **Timer** *Trigger Type* but not both as they are mutually exclusive fields
- For a **Compensation** *Intermediate Event* edge-mounted on an *Activity*, create a BPMN 1.1 *Association* from this event to the *Compensation Activity*; ensure that the *IsCompensation* tag for the *Activity* is set to true
- **Link** *Intermediate Event* can be used either as a GOTO or an off-page connector; therefore, this event can have either incoming or outgoing *Sequence Flows* - but not both

8.3.5.1.2.3 Activity

A BPMN 1.1 *Activity* represents work that is performed within a *Process*. An *Activity* can be modeled as a:

- **Sub-Process** - a compound *Activity* that is defined as a flow of other **BPMN 1.1** elements or
- **Task** - an atomic *Activity* that cannot be broken down into a smaller unit

Activities - both *Tasks* and *Sub-Processes* - can also, optionally, act as *Looping constructs*. The *OMG BPMN 1.1 Specification* defines two types of *Looping construct*:

- **Standard** Loop (*while* or *until*)
- **Multi-Instance** Loop (*for each*)

BPEL Properties:


- Double-click on the *Activity* in the BPEL diagram
- Right-click on the *Activity* | **BPEL | BPEL 1.1 Properties**

Reference

Field/Button	Usage	See also
Name	Specify the name for the <i>Activity</i> .	
Type	Specify whether the <i>Activity</i> is a : <ul style="list-style-type: none"> • Task (or) • Sub-Process 	
Task Type / SubProcess Type	Depending on the value selected in the Type field, the Task Type / SubProcess Type field has the following values: <p>Task Type :</p> <ul style="list-style-type: none"> • Manual 	

	<ul style="list-style-type: none"> • None • Receive • Reference • Script • Send • Service • User <p>SubProcess Type :</p> <ul style="list-style-type: none"> • Embedded • References • Reusable 																							
<p>Details</p>	<p>Depending on the selected Task Type, the Details tab changes as follows:</p> <table border="1" data-bbox="496 1003 1214 2042"> <thead> <tr> <th data-bbox="496 1003 703 1061">Field</th> <th data-bbox="703 1003 1214 1061">Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2" data-bbox="496 1061 1214 1115">Receive Task Type</td> </tr> <tr> <td data-bbox="496 1115 703 1272">Web Service</td> <td data-bbox="703 1115 1214 1272">Select a BPMN 1.1 <i>Web Service</i> from the pack (the <i>SupportingElements</i> package) that repres of the involved <i>Participants</i>.</td> </tr> <tr> <td data-bbox="496 1272 703 1406">Message</td> <td data-bbox="703 1272 1214 1406">Select a BPMN 1.1 <i>Message</i> from the list of M in the selected Web Service.</td> </tr> <tr> <td data-bbox="496 1406 703 1541">Instantiate</td> <td data-bbox="703 1406 1214 1541">Select True if this is the first <i>Activity</i> after the S <input data-bbox="715 1451 738 1485" type="checkbox"/>, otherwise select False.</td> </tr> <tr> <td colspan="2" data-bbox="496 1541 1214 1594"></td> </tr> <tr> <td colspan="2" data-bbox="496 1594 1214 1648">Reference Task Type</td> </tr> <tr> <td data-bbox="496 1648 703 1783">Activity</td> <td data-bbox="703 1648 1214 1783">Select a BPMN 1.1 <i>Task</i> from the list of all Act the <i>Process</i>.</td> </tr> <tr> <td colspan="2" data-bbox="496 1783 1214 1836"></td> </tr> <tr> <td colspan="2" data-bbox="496 1836 1214 1890">Send Task Type</td> </tr> <tr> <td data-bbox="496 1890 703 2042">Web Service</td> <td data-bbox="703 1890 1214 2042">Select a BPMN 1.1 <i>Web Service</i> from the pack (the <i>SupportingElements</i> package) that repres of the involved <i>Participants</i>.</td> </tr> </tbody> </table>	Field	Usage	Receive Task Type		Web Service	Select a BPMN 1.1 <i>Web Service</i> from the pack (the <i>SupportingElements</i> package) that repres of the involved <i>Participants</i> .	Message	Select a BPMN 1.1 <i>Message</i> from the list of M in the selected Web Service.	Instantiate	Select True if this is the first <i>Activity</i> after the S <input data-bbox="715 1451 738 1485" type="checkbox"/> , otherwise select False .			Reference Task Type		Activity	Select a BPMN 1.1 <i>Task</i> from the list of all Act the <i>Process</i> .			Send Task Type		Web Service	Select a BPMN 1.1 <i>Web Service</i> from the pack (the <i>SupportingElements</i> package) that repres of the involved <i>Participants</i> .	
Field	Usage																							
Receive Task Type																								
Web Service	Select a BPMN 1.1 <i>Web Service</i> from the pack (the <i>SupportingElements</i> package) that repres of the involved <i>Participants</i> .																							
Message	Select a BPMN 1.1 <i>Message</i> from the list of M in the selected Web Service.																							
Instantiate	Select True if this is the first <i>Activity</i> after the S <input data-bbox="715 1451 738 1485" type="checkbox"/> , otherwise select False .																							
Reference Task Type																								
Activity	Select a BPMN 1.1 <i>Task</i> from the list of all Act the <i>Process</i> .																							
Send Task Type																								
Web Service	Select a BPMN 1.1 <i>Web Service</i> from the pack (the <i>SupportingElements</i> package) that repres of the involved <i>Participants</i> .																							

	<table border="1"> <tr> <td></td> <td></td> </tr> <tr> <td>Message</td> <td>Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected Web Service.</td> </tr> <tr> <td colspan="2">Service / User Task Type</td> </tr> <tr> <td>Web Service</td> <td>Select a BPMN 1.1 <i>Web Service</i> from the pack (the <i>SupportingElements</i> package) that repres of the involved <i>Participants</i>.</td> </tr> <tr> <td>Input Message</td> <td>Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected Web Service that represents th element.</td> </tr> <tr> <td>Output Message</td> <td>Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected Web Service that represents th element.</td> </tr> <tr> <td colspan="2">Depending on the selected SubProcess Type, the Details tab changes as follows :</td> </tr> <tr> <td>Field</td> <td>Usage</td> </tr> <tr> <td colspan="2">References SubProcess Type</td> </tr> <tr> <td>SubProcess</td> <td>Select a BPMN 1.1 <i>Sub-Process</i> from the li <i>Activities</i> in the <i>Process</i>.</td> </tr> </table>			Message	Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected Web Service.	Service / User Task Type		Web Service	Select a BPMN 1.1 <i>Web Service</i> from the pack (the <i>SupportingElements</i> package) that repres of the involved <i>Participants</i> .	Input Message	Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected Web Service that represents th element.	Output Message	Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected Web Service that represents th element.	Depending on the selected SubProcess Type , the Details tab changes as follows :		Field	Usage	References SubProcess Type		SubProcess	Select a BPMN 1.1 <i>Sub-Process</i> from the li <i>Activities</i> in the <i>Process</i> .	
Message	Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected Web Service.																					
Service / User Task Type																						
Web Service	Select a BPMN 1.1 <i>Web Service</i> from the pack (the <i>SupportingElements</i> package) that repres of the involved <i>Participants</i> .																					
Input Message	Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected Web Service that represents th element.																					
Output Message	Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected Web Service that represents th element.																					
Depending on the selected SubProcess Type , the Details tab changes as follows :																						
Field	Usage																					
References SubProcess Type																						
SubProcess	Select a BPMN 1.1 <i>Sub-Process</i> from the li <i>Activities</i> in the <i>Process</i> .																					
Assignments	Select one or more <i>Assignment</i> elements created in the <i>SupportingElements</i> package in this tab (Optional).	Assignment ^[1589] Create BPEL 1.1 Model Structure <small>[1573]</small>																				
Loop Details	Activities can be repeated sequentially, behaving like a loop. Specify the <i>Activity</i> looping details in this tab (Optional).																					
	<table border="1"> <tr> <td>Field</td> <td>Usage</td> </tr> <tr> <td>Type</td> <td> Select: <ul style="list-style-type: none"> • Standard for creating a <i>while</i> or <i>until</i> • Multinstance for creating a <i>for each</i> </td> </tr> </table>	Field	Usage	Type	Select: <ul style="list-style-type: none"> • Standard for creating a <i>while</i> or <i>until</i> • Multinstance for creating a <i>for each</i> 																	
Field	Usage																					
Type	Select: <ul style="list-style-type: none"> • Standard for creating a <i>while</i> or <i>until</i> • Multinstance for creating a <i>for each</i> 																					

	Condition	Enter a boolean condition for Standard loop numeric condition for MultiInstance loop.
	Min Value	Specify the minimum value for the loop even a Standard loop(Optional).
	Min Value	Specify the maximum value for the loop even a Standard loop(Optional).
	Test Time	Select After to define a <i>while</i> Standard loop Before to define an <i>until</i> Standard loop.
OK	Save the values entered in the dialog.	
Cancel	Discard the values entered in the dialog.	
Help	Display this Help topic.	
General	Open the UML Properties dialog.	Properties 

Notes

- **Manual** and **Script** types cannot be mapped to BPEL 1.1
- **Reusable Sub-Process** types cannot be mapped to BPEL 1.1

8.3.5.1.2.4 Gateway

A BPMN 1.1 *Gateway* control the way in which *Sequence Flows* converge and diverge within a *Process*. They provide a gating mechanism that either allows or blocks a *Sequence Flow*.

BPEL Properties:

- Double-click on the *Gateway* in the BPEL diagram
- Right-click on the *Gateway* | **BPEL** | **BPEL Properties**

Reference

Field/Button	Usage	See also
Name	Specify the name for the <i>Gateway</i> .	

Gateway	Select the type for the <i>Gateway</i> , namely: <ul style="list-style-type: none"> • Complex • Exclusive • Inclusive • Parallel 							
Details	Depending on the selected Gateway , the Details tab changes as follows: <table border="1" data-bbox="513 667 1295 990"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Exclusive Gateway</td> </tr> <tr> <td>Exclusive Type</td> <td> Data - allows alternate paths to be taken based on the evaluation of conditions. Event - allows alternate paths to be taken based on the occurrence of events. </td> </tr> </tbody> </table>	Field	Usage	Exclusive Gateway		Exclusive Type	Data - allows alternate paths to be taken based on the evaluation of conditions. Event - allows alternate paths to be taken based on the occurrence of events.	
Field	Usage							
Exclusive Gateway								
Exclusive Type	Data - allows alternate paths to be taken based on the evaluation of conditions. Event - allows alternate paths to be taken based on the occurrence of events.							
OK	Save the values entered in the dialog.							
Cancel	Discard the values entered in the dialog.							
Help	Display this Help topic.							
General	Open the UML Properties dialog.	Properties 8431						

Notes

- The target of the outgoing *Sequence Flows* of this Event *Exclusive Gateway* must be either a:
 - **Receive Task** or a
 - **Message** or **Timer Intermediate Event**
- If a *Message Task* is one of the targets of the outgoing *Sequence Flow* of an Event *Exclusive Gateway*, then a *Message Intermediate Event* cannot be the target of the other outgoing *Sequence Flows* of this Event *Exclusive Gateway*

8.3.5.1.2.5 End Event

A BPMN 1.1 *End Event* indicates where a particular *Process* ends. Every *Process* in Enterprise Architect must end with an *End Event*.

BPEL Properties:

- Double-click on the *End Event* in the BPEL diagram
- Right-click on the *End Event* | **BPEL** | **BPEL Properties**

Reference

Field/Button	Usage	See also										
Name	Specify the name for the <i>End Event</i> .											
Result Type	Select the type for the <i>End Event</i> , namely: <ul style="list-style-type: none"> • Cancel • Compensation • Error • Link • Message • Multiple • None • Signal • Terminate 											
Details	Depending on the selected Result Type , the Details tab changes as follows: <table border="1" style="margin-top: 10px; width: 100%;"> <thead> <tr> <th style="background-color: #e0e0e0;">Field</th> <th style="background-color: #e0e0e0;">Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Compensation Result Type</td> </tr> <tr> <td>Activity</td> <td>Select a BPMN 1.1 <i>Task</i> from the list of all <i>Act</i> the <i>Process</i>.</td> </tr> <tr> <td colspan="2">Error Trigger Type</td> </tr> <tr> <td>Error Code</td> <td>Specify the required error code.</td> </tr> </tbody> </table>	Field	Usage	Compensation Result Type		Activity	Select a BPMN 1.1 <i>Task</i> from the list of all <i>Act</i> the <i>Process</i> .	Error Trigger Type		Error Code	Specify the required error code.	
Field	Usage											
Compensation Result Type												
Activity	Select a BPMN 1.1 <i>Task</i> from the list of all <i>Act</i> the <i>Process</i> .											
Error Trigger Type												
Error Code	Specify the required error code.											

	<table border="1"> <tr> <td colspan="2">Message Result Type</td> </tr> <tr> <td>Web Service</td> <td>Select a BPMN 1.1 <i>Web Service</i> from the pack (the <i>SupportingElements</i> package) that represents the involved <i>Participants</i>.</td> </tr> <tr> <td>Message</td> <td>Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected <i>Web Service</i>.</td> </tr> <tr> <td colspan="2">Multiple Result Type</td> </tr> <tr> <td>Events</td> <td>Select additional BPMN 1.1 <i>Start Events</i> from events in this <i>Process</i> that might trigger the <i>P</i> clicking the (...) button.</td> </tr> </table>	Message Result Type		Web Service	Select a BPMN 1.1 <i>Web Service</i> from the pack (the <i>SupportingElements</i> package) that represents the involved <i>Participants</i> .	Message	Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected <i>Web Service</i> .	Multiple Result Type		Events	Select additional BPMN 1.1 <i>Start Events</i> from events in this <i>Process</i> that might trigger the <i>P</i> clicking the (...) button.	
Message Result Type												
Web Service	Select a BPMN 1.1 <i>Web Service</i> from the pack (the <i>SupportingElements</i> package) that represents the involved <i>Participants</i> .											
Message	Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected <i>Web Service</i> .											
Multiple Result Type												
Events	Select additional BPMN 1.1 <i>Start Events</i> from events in this <i>Process</i> that might trigger the <i>P</i> clicking the (...) button.											
Assignments	Select one or more <i>Assignment</i> elements created in the <i>SupportingElements</i> package in this tab (Optional).	Assignment ¹⁵⁸⁹ Create BPEL 1.1 Model Structure ¹⁵⁷³										
OK	Save the values entered in the dialog.											
Cancel	Discard the values entered in the dialog.											
Help	Display this Help topic.											
General	Open the UML Properties dialog.	Properties ⁸⁴³										

Notes

- **Cancel**, **Link**, **None** and **Signal Event** types cannot be mapped to BPEL 1.1

8.3.5.1.2.6 Sequence Flow

A BPMN 1.1 *Sequence Flow* connector shows the order in which the activities (*Tasks* and *Events*) are performed in a *BPEL Process*.

BPEL Properties:

- Double-click on the *Sequence Flow* in the BPEL diagram
- Right-click on the *Sequence Flow* | **BPEL** | **BPEL Properties**

Reference

Field/Button	Usage	See also
Condition Type	Specify the type of the condition on the <i>Sequence Flow</i> , namely : <ul style="list-style-type: none"> • None • Default • Expression 	
Expression	This field gets enabled when the Condition Type is set to Expression . Specify a boolean expression to act as a gating condition.	
Ordering	This field gets enabled when the Condition Type is set to Expression . Specify a numerical value that determines the order in which the condition set in the Expression field is to be evaluated.	
Assignments	Select one or more <i>Assignment</i> elements created in the <i>SupportingElements</i> package in this tab (Optional).	Assignment ^[1589] Create BPEL 1.1 Model Structure ^[1573]
OK	Save the values entered in the dialog.	
Cancel	Discard the values entered in the dialog.	
Help	Display this Help topic.	
General	Open the UML Properties dialog.	Properties ^[843]

8.3.5.1.2.7 Pool

A BPMN 1.1 *Pool* represents a Participant in a *Process* and does not map to any specific BPEL 1.1 element. Enterprise Architect uses Pools to represent external Participants, with which the BPEL Process communicates. These are 'black box' *pools*; that is, they are abstract and do not expose any details (they do not contain any BPMN 1.1 elements inside them).

BPEL Properties:

- Double-click on the *Pool* in the BPEL diagram
- Right-click on the *Pool* | **BPEL** | **BPEL Properties**

Reference

Field/Button	Usage	See also
Name	Specify the name for the <i>Pool</i> .	
OK	Save the values entered in the dialog.	
Cancel	Discard the values entered in the dialog.	
Help	Display this Help topic.	
General	Open the UML Properties dialog.	Properties [843]

Notes

- A *BPEL Process* should not contain a *Pool* as its child element
- A *BPEL* diagram under a *BPEL Process* contains an implicit *Pool* - so it is invalid to add a *Pool* in this *BPEL* diagram to represent the *BPEL Process*
- *Pool* cannot have any incoming or outgoing *Sequence Flow* connectors - it can have only incoming or outgoing *Message Flow* connectors

8.3.5.1.2.8 Assignment

A BPMN 1.1 *Assignment* element enables data to be copied between messages and new data to be inserted, using expressions within a *BPEL Process*. A BPMN 1.1 *Assignment* element maps to a BPEL 1.1 *assign* activity and copies the specified value from the source to the target.

In Enterprise Architect, *Assignment* elements should be created in the *Assignments* package in *SupportingElements*^[1598]. If they are created elsewhere, they cannot be enacted correctly.

Access: [View](#) | [Diagram Toolbox: More tools](#) | [BPMN 1.1](#) | [BPMN 1.1 Types](#)

BPEL Properties:

- Double-click on the *Assignment* in the *BPEL* diagram
- Right-click on the *Assignment* | [BPEL](#) | [BPEL Properties](#)

Reference

Field/Button	Usage	See also
Name	Specify the name for the <i>Assignment</i> .	

Assign Time	Select either Start or End to determine whether the assignment occurs at the start or end of an <i>Activity/Event</i> .	
Copy From section		
Type	Depending on the value selected in this field, further details are required.	
Literal	Specify a literal value.	
Expression / Message	Select a BPMN 1.1 <i>Message</i> in the package representing the <i>BPEL Process / Pool</i> under the <i>SupportingElements</i> package.	Create BPEL 1.1 Model Structure ^[1573] Create a BPEL 1.1 Web Service ^[1591]
Part	Select a BPMN 1.1 <i>Property</i> belonging to the selected Message (Optional).	
Copy To section		
Message	Select a BPMN 1.1 <i>Message</i> in the package representing the <i>BPEL Process / Pool</i> under the <i>SupportingElements</i> package.	Create BPEL 1.1 Model Structure ^[1573] Create a BPEL 1.1 Web Service ^[1591]
Part	Select a BPMN 1.1 <i>Property</i> belonging to the selected Message (Optional).	
OK	Save the values entered in the dialog.	
Cancel	Discard the values entered in the dialog.	
Help	Display this Help topic.	
General	Open the UML Properties dialog.	Properties ^[843]

Notes

- *Messages* are created when you create Web Services
- If you select **Expression**, Enterprise Architect uses the *getVariableData XPATH 1.0* function to create the expression from the selected Message and Part

8.3.5.1.3 Create BPEL 1.1 Web Service

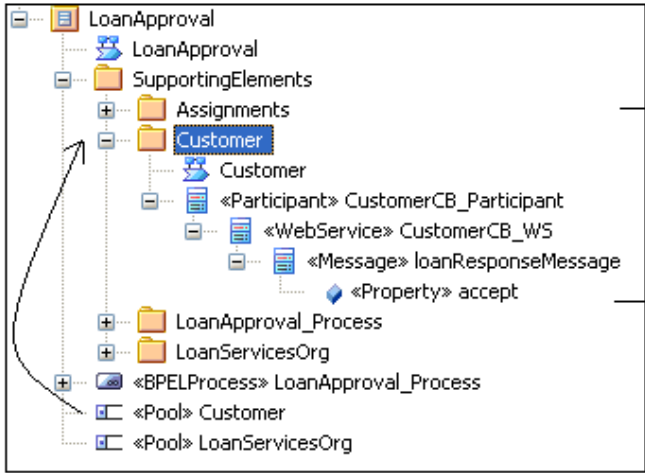
BPEL is an **Orchestration** language, which orchestrates services that are exposed using **WSDL 1.1**. It coordinates the execution of the various operations of these involved services. BPEL 1.1 supports **WSDL 1.1 one-way** and **request-response** operations only (and not **solicit-response** and **notification** operations). Enterprise Architect enables you to create, for *BPEL Process* and *Pool* elements, **Web Service Operations** that support either *synchronous* (request-response) or *asynchronous* (one-way) interactions.

Access Right-click on a *BPEL Process* or *Pool* | **BPEL | Create WebService**

Reference

Field/Button	Usage	See also						
Operation	Specify whether to create a new Web Service or create one from an existing WSDL created / imported into Enterprise Architect (using the Import WSDL dialog) by selecting either : <ul style="list-style-type: none"> • Create New (or) • Create from existing WSDL 	Model WSDL <small>[2069]</small> Import WSDL <small>[2086]</small>						
Type	Specify whether to create a <i>one-way</i> or <i>request-response</i> Operation by selecting either : <ul style="list-style-type: none"> • Asynchronous (or) • Synchronous 							
Partnership Details	<p>The interaction between the Web Service and the <i>BPEL Process</i> is modeled as a BPEL <i>partnerLink</i>. For this interaction, provide a:</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Name of the BPEL <i>partnerLink</i>.</td> </tr> <tr> <td>Role</td> <td>Name of either: <ul style="list-style-type: none"> • <i>myRole</i> (if this operation belongs to <i>BPEL Process</i> Web Service) or • <i>partnerRole</i> (if this operation belongs to Web Service of a <i>Pool</i>) of the BPEL <i>partnerLink</i>. </td> </tr> </tbody> </table>	Field	Usage	Name	Name of the BPEL <i>partnerLink</i> .	Role	Name of either: <ul style="list-style-type: none"> • <i>myRole</i> (if this operation belongs to <i>BPEL Process</i> Web Service) or • <i>partnerRole</i> (if this operation belongs to Web Service of a <i>Pool</i>) of the BPEL <i>partnerLink</i>. 	
Field	Usage							
Name	Name of the BPEL <i>partnerLink</i> .							
Role	Name of either: <ul style="list-style-type: none"> • <i>myRole</i> (if this operation belongs to <i>BPEL Process</i> Web Service) or • <i>partnerRole</i> (if this operation belongs to Web Service of a <i>Pool</i>) of the BPEL <i>partnerLink</i>. 							

Web Service	Field Usage	
	When Create New is selected in the Operation field:	
	Web Service Name	Name of the WSDL 1.1 Web Service.
	PortType	Name of the WSDL 1.1 <i>PortType</i> (Interface)
	Operation	Name of the WSDL 1.1 <i>PortType</i> <i>Operation</i> .
	When Create from existing WSDL PortType Operation is selected the Operation field:	
	WSDL Package	Select an existing WSDL package created / imported into Enterprise Architect.
	Web Service Name	Name of the WSDL 1.1 Web Service. Default: name of the selected WSDL Package
	PortType	Select a WSDL 1.1 <i>PortType</i> from the selected WSDL Package.
	Operation	Select a WSDL 1.1 <i>PortType</i> <i>Operation</i> in the selected PortType .
Input	Field Usage	
	When Create New is selected in the Operation field:	
	Message Name	Name of the WSDL 1.1 <i>Message</i> .
	Properties	Press (...) to enter the WSDL 1.1 <i>Message Name</i> and XSD Type .
	When Create from existing WSDL PortType Operation is selected the Operation field: The fields in this tab are pre-filled with the details of the input WSDL <i>Message</i> (of the WSDL 1.1 <i>PortType</i> <i>Operation</i> selected in the Operation field in the Web Service tab).	

<p>Output</p>	<table border="1"> <thead> <tr> <th data-bbox="496 383 735 439">Field</th> <th data-bbox="735 383 1222 439">Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2" data-bbox="496 450 1222 495">When Create New is selected in the Operation field:</td> </tr> <tr> <td data-bbox="496 495 735 595">Message Name</td> <td data-bbox="735 495 1222 595">The name of the WSDL 1.1 <i>Message</i>.</td> </tr> <tr> <td data-bbox="496 595 735 730">Properties</td> <td data-bbox="735 595 1222 730">Press the (...) button to enter the WSDL 1.1 <i>Message Part Name</i> and XSD Type.</td> </tr> <tr> <td colspan="2" data-bbox="496 786 1222 864">When Create from existing WSDL PortType Operation is selected in the Operation field:</td> </tr> <tr> <td colspan="2" data-bbox="496 864 1222 1021">The fields in this tab are pre-filled with the details of the output <i>WS Message</i> (of the WSDL 1.1 <i>PortType Operation</i> selected in the Operation field in the Web Service tab).</td> </tr> </tbody> </table>	Field	Usage	When Create New is selected in the Operation field:		Message Name	The name of the WSDL 1.1 <i>Message</i> .	Properties	Press the (...) button to enter the WSDL 1.1 <i>Message Part Name</i> and XSD Type .	When Create from existing WSDL PortType Operation is selected in the Operation field:		The fields in this tab are pre-filled with the details of the output <i>WS Message</i> (of the WSDL 1.1 <i>PortType Operation</i> selected in the Operation field in the Web Service tab).		
Field	Usage													
When Create New is selected in the Operation field:														
Message Name	The name of the WSDL 1.1 <i>Message</i> .													
Properties	Press the (...) button to enter the WSDL 1.1 <i>Message Part Name</i> and XSD Type .													
When Create from existing WSDL PortType Operation is selected in the Operation field:														
The fields in this tab are pre-filled with the details of the output <i>WS Message</i> (of the WSDL 1.1 <i>PortType Operation</i> selected in the Operation field in the Web Service tab).														
<p>OK</p>	<p>Create a Web Service Operation based on the values entered in the dialog.</p> <p>The Operation is created in a package, whose name is the same as that of the <i>BPEL Process / Pool</i> from which this dialog is invoked, under the <i>SupportingElements</i> package.</p> 	<p>SupportingElements ¹⁵⁹³</p>												

Cancel	Discard the values entered in the dialog and abort creating the Web Service Operation.	
Help	Display this Help topic.	

Notes

- The Output tab is not applicable for an Asynchronous operation

8.3.5.1.4 Generate BPEL 1.1

BPEL 1.1 code can be generated from a *BPEL Process*. Enterprise Architect validates the *BPEL Process* before generating the **BPEL 1.1** code. In addition to generating the **BPEL 1.1** code, **WSDL 1.1** files are generated for the *BPEL Process* and all the involved *Pools* (provided **Web Services** are defined for them).

Access Right-click on a BPEL Process and select the following context menu: **BPEL | Generate BPEL**

Reference

Field/Button	Usage	See also
File Name	Specify the path where the BPEL 1.1 file is to be generated.	
Namespace Details	Double-click on an entry (if any) in this field to open the Namespace Details dialog and add / edit the namespace details. The entry DefaultPool represents the current <i>BPEL Process</i> .	
Generate BPEL	Validate the model and generate BPEL 1.1.	BPEL Model Validation ^[1615]
Close	Close this dialog.	
Help	Display this Help topic.	
View BPEL	View the generated BPEL 1.1 file.	

Learn more

- [Model a BPEL 1.1 Process](#)^[1578]
- [Create a BPEL 1.1 Web Service](#)^[1591]

8.3.5.2 BPEL 2.0 Model

Enterprise Architect uses BPMN 2.0 as a graphical front-end to capture BPEL 2.0 *Process* description (as BPEL 2.0 does not have not a standard graphical notation). Enterprise Architect uses the partial mapping specified in the BPMN 2.0 specification to map BPMN 2.0 constructs to BPEL 2.0.

How to

Step	Action	See also										
1	<p>Create a BPEL 2.0 Package Structure.</p> <p>Alternatively:</p> <table border="1"> <thead> <tr> <th>Step</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>a</td> <td>Create a package in the Project Browser.</td> </tr> <tr> <td>b</td> <td>Click on the New Diagram icon in the Project Browser toolbar.</td> </tr> <tr> <td>c</td> <td>Select BPMN 2.0 in the Select From field and BPEL in the Diagram</td> </tr> <tr> <td>d</td> <td>Click on the BPEL icon from the BPMN 2.0 - Business Process To into the created BPEL diagram.</td> </tr> </tbody> </table>	Step	Action	a	Create a package in the Project Browser.	b	Click on the New Diagram icon in the Project Browser toolbar.	c	Select BPMN 2.0 in the Select From field and BPEL in the Diagram	d	Click on the BPEL icon from the BPMN 2.0 - Business Process To into the created BPEL diagram.	<p>BPEL 2.0 Package Structure^[1597]</p> <p>Project Browser Toolbar^[583]</p>
Step	Action											
a	Create a package in the Project Browser.											
b	Click on the New Diagram icon in the Project Browser toolbar.											
c	Select BPMN 2.0 in the Select From field and BPEL in the Diagram											
d	Click on the BPEL icon from the BPMN 2.0 - Business Process To into the created BPEL diagram.											
2	Open the BPEL diagram under the <i>BPELProcess</i> stereotyped element.											
3	<p>Model the <i>BPEL Process</i> using the constructs in the BPMN 2.0 - Business Process Toolbox. Use the following constructs in this toolbox to model BPEL 2.0:</p> <p>BPMN 2.0 - Business Process page:</p> <ul style="list-style-type: none"> • BPEL • Activity • Data Object • Start Event • Intermediate Event • Event • Gateway • Pool <p>BPMN 2.0 - Business Process Connectors page:</p> <ul style="list-style-type: none"> • Sequence Flow 	<p>BPMN 2.0 Business Process Toolbox^[1550]</p> <p>Model BPEL 2.0 Process^[1599]</p>										

	<ul style="list-style-type: none"> • Association • Message Flow <p>BPMN 2.0 Types page:</p> <ul style="list-style-type: none"> • Assignment • Error • Interface • Operation • Participant • Property 	
4	Create Web Service Operations for the <i>BPEL Process</i> and other related <i>Pools</i> , if required.	Web Service Operation ^[1615]
5	Generate BPEL 2.0 code from the <i>BPEL Process</i> .	Generate BPEL 2.0 ^[1618]

Notes

- Generating BPEL 2.0 from a BPMN 2.0 model is supported in the Business and Software Engineering and the Ultimate editions of Enterprise Architect
- Ensure that BPMN 2.0 Technology is enabled in the MDG Technologies dialog (**Settings | MDG Technologies**)

BPEL 2.0 Example:

- The Enterprise Architect Example file (*EAExample.EAP*) has a sample BPMN 2.0 model from which BPEL 2.0 can be generated
- If you have installed Enterprise Architect at the default location, open this file: *C:\Program Files\Sparx Systems\EA\EAExample.EAP*
- The BPMN 2.0 model package is in: **Project Models | Analysis and Business Modeling | BPEL Example | BPEL 2.0 Model**

Modeling Restrictions:

- Every *BPEL Process* and *Sub-Process* should start with a *Start Event* and end with an *End Event*
 - *Boundary Start and End Events* are not supported
- *Sequence Flow* Looping is not supported in both **Normal** and **Exception Paths**
 - All *Sequence Flows* should flow downstream and not upstream

- *Sub-Process* cannot be a **Loop** node and have boundary *Intermediate Events*
- *Event Sub-Process* cannot act as a **Loop** Node
- *Assignments* are not supported on:
 - *Start Events* on *Event Sub-Process*
 - *End Events*
 - *Sub-Process*
 - Boundary *Intermediate Event*
 - *Gateway*
 - *Task* and *Intermediate Event* that immediately follow an *XOR Event Gateway*
- **Exception Path** has to merge back into the **Normal Path**
 - An exception to this rule is boundary *Compensation Intermediate Event* which should have a BPMN 2.0 *Association* to a *Compensation Activity* (that has no incoming or outgoing *Sequence Flows*)
 - Multiple **Exception Paths** from an *Activity* must join at the same location in the **Normal Path**
 - An **Exception Path** should not cross another **Exception Path**
- *Activities* in an **Exception Path** cannot have boundary **Intermediate Events**.
- *Pools* are treated as *black boxes* (that is, they do not expose any details) and hence they cannot contain any child elements or have any incoming/outgoing *Sequence Flow* connectors

Learning Center topics

- **(Alt + F1) | Business Modeling | BPEL | Getting Started**

8.3.5.2.1 Create BPEL 2.0 Model Structure

A BPEL 2.0 model consists of a *BPEL Process* (containing a BPEL diagram and mappable BPMN 2.0 constructs) and other supporting elements (like BPMN 2.0 *Assignment*, BPMN 2.0 *Operation*) required for generating a BPEL 2.0 code.

A sample BPEL 2.0 package structure can be created in the Project Browser, using the Select Model(s) (Model Wizard) dialog. You can use this package structure as a template for developing your *BPEL Process*.

Access

- **Project Browser package context menu | Add | Add a New Model using Wizard**, or
- **Project Browser toolbar: New Model from Pattern**

Use to

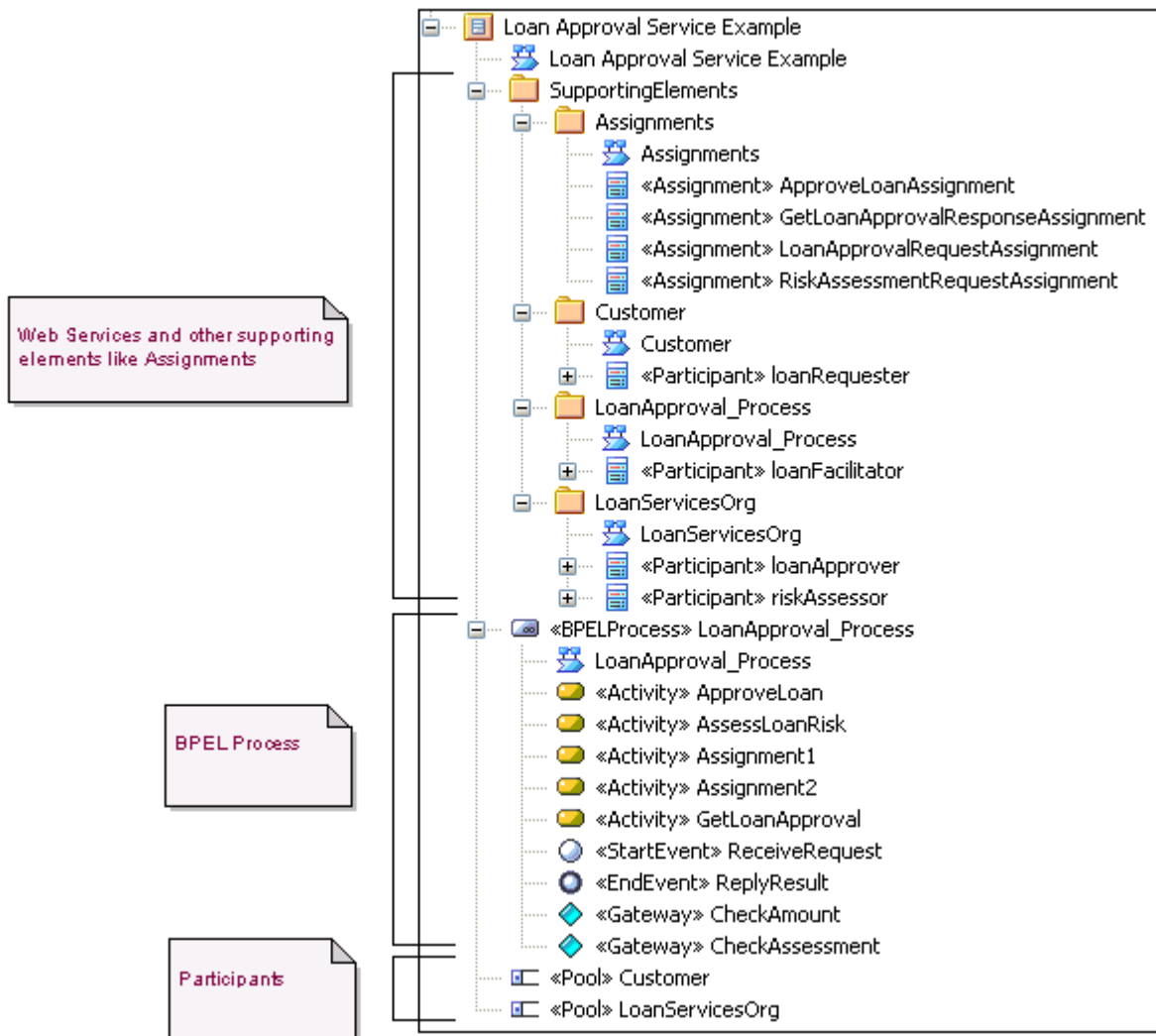
- Create a new BPEL 2.0 package structure that can be used as a starting point for developing your

*BPEL Process***How to**

Step	Action	See also
1	Select the root node or a package in the Project Browser.	
2	Click on the New Model from Pattern icon in the Project Browser toolbar.	Project Browser Toolbar [583]
3	Select the value BPMN 2.0 in the Technology section.	Model Wizard [660]
4	Check the option BPEL 2.0 Model in the Name section.	
5	Click on the OK button to create the sample BPEL 2.0 package structure.	

Example BPEL 2.0 Package Structure:

The *BPEL Process* **LoanApproval_Process** acts as container for the BPEL diagram and elements. The *SupportingElements* package contains supporting components like **Assignments** and **Web Service Operations**.



Learn more

- [Model a BPEL 2.0 Process](#) ^[1599]
- [Create a Web Service Operation](#) ^[1615]
- [Generate BPEL 2.0](#) ^[1618]

8.3.5.2.2 Model a BPEL 2.0 Process

The *BPEL Process* in Enterprise Architect represents the top-level container for the **BPMN 2.0** elements, from which **BPEL 2.0** can be generated. Conceptually it maps to the BPEL *process* element.

BPEL Properties:

- Double-click on the *BPEL Process* in the **BPEL** diagram (or)
- Right-click on the *BPEL Process* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL 2.0 Properties**

Reference

Field/Button	Usage	See also
Name	Specify the name for the <i>BPEL Process</i> .	
Query Language	Specify the value of the language used in the <i>BPEL Process</i> for the selection of nodes in <i>Assignments</i> . Defaults to XPath 1.0 .	
OK	Save the values entered in the dialog.	
Cancel	Discard the values entered in the dialog.	
Help	Display this Help topic.	
General	Open the UML Properties dialog.	Properties [843]

Learn more

The *BPEL Process* element acts as a container for the following BPMN 2.0 constructs:

- [Start Event](#) [1601]
- [Intermediate Event](#) [1602]
- [Activity](#) [1605]
- [Gateway](#) [1607]
- [End Event](#) [1608]
- [Data Object](#) [1610]
- [Property](#) [1611]
- [Sequence Flow](#) [1612]

The following BPMN 2.0 constructs can be optionally used in BPEL modeling:

- [Pool](#) [1588]
- [Assignment](#) [1589]

Learning Center topics

- **(Alt + F1) | Business Modeling | BPEL | Process Modeling**

8.3.5.2.2.1 Start Event

A BPMN 2.0 *Start Event* indicates where a particular *Process* begins. Every *Process* in Enterprise Architect must begin with a *Start Event*.

BPEL Properties:

- Double-click on the *Start Event* in the BPEL diagram, or
- Right-click on the *Start Event* | **BPEL** | **BPEL 2.0 Properties**

Reference

Field/Button	Usage	See also										
Name	Specify the name for the <i>Start Event</i> .											
Event Type	Select a trigger for the <i>Start Event</i> , namely: <ul style="list-style-type: none"> • Compensation • Conditional • Error • Escalation • Message • Multiple • None • Parallel Multiple • Signal • Timer 											
Details	Depending on the selected Event Type , the Details tab changes as follows: <table border="1" data-bbox="491 1594 1220 2027"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Message Event Type</td> </tr> <tr> <td>Pool Package</td> <td>Select the package that represents one of the involved <i>Participants</i>.</td> </tr> <tr> <td>Operation</td> <td>Select an Operation from the list of operations selected Pool Package.</td> </tr> <tr> <td></td> <td></td> </tr> </tbody> </table>	Field	Usage	Message Event Type		Pool Package	Select the package that represents one of the involved <i>Participants</i> .	Operation	Select an Operation from the list of operations selected Pool Package .			
Field	Usage											
Message Event Type												
Pool Package	Select the package that represents one of the involved <i>Participants</i> .											
Operation	Select an Operation from the list of operations selected Pool Package .											

	<table border="1"> <tr> <td colspan="2">Timer Event Type</td> </tr> <tr> <td>Time Cycle</td> <td>Specify the value for the time duration.</td> </tr> <tr> <td>Time Date</td> <td>Specify the value for the time date.</td> </tr> <tr> <td colspan="2"> </td> </tr> <tr> <td colspan="2">Error Event Type</td> </tr> <tr> <td>Error</td> <td>Select an <i>Error</i> element.</td> </tr> <tr> <td colspan="2"> </td> </tr> </table>	Timer Event Type		Time Cycle	Specify the value for the time duration.	Time Date	Specify the value for the time date.			Error Event Type		Error	Select an <i>Error</i> element.			
Timer Event Type																
Time Cycle	Specify the value for the time duration.															
Time Date	Specify the value for the time date.															
Error Event Type																
Error	Select an <i>Error</i> element.															
Assignments	Select one or more <i>Assignment</i> elements created in the <i>SupportingElements</i> package in this tab (Optional).	Assignment ^[1613] SupportingElements ^[1598]														
OK	Save the values entered in the dialog.															
Cancel	Discard the values entered in the dialog.															
Help	Display this Help topic.															
General	Open the UML Properties dialog.	Properties ^[843]														

Notes

- Either set **Time Cycle** or **Time Date** for **Timer Event Type** but not both, as they are mutually exclusive fields
- The Assignments tab is not available on *Start Events* that are used to start an *Event Sub-Process*
- **Compensation**, **Error** and **Timer** event types are valid only on an *Event Sub-Process*
- **Conditional**, **Escalation**, **Multiple**, **Parallel Multiple** and **Signal** event types cannot be mapped to BPEL 2.0

8.3.5.2.2 Intermediate Event

A BPMN 2.0 *Intermediate Event* indicates where an event occurs somewhere between the start and end of a *Process*.

BPEL Properties:

- Double-click on the *Intermediate Event* in the BPEL diagram
- Right-click on the *Intermediate Event* | **BPEL** | **BPEL 2.0 Properties**

Reference

Field/Button	Usage	See also												
Name	Specify the name for the <i>Intermediate Event</i>													
Event Type	<p>Select the type for the <i>Intermediate Event</i>, namely:</p> <ul style="list-style-type: none"> • Cancel • Compensation • Conditional • Error • Escalation • Link • Message • Multiple • None • Parallel Multiple • Signal • Timer 													
Details	<p>Depending on the selected Event Type, the Details tab changes as follows:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Message Event Type</td> </tr> <tr> <td>Pool Package</td> <td>Select the package that represents one of involved <i>Participants</i>.</td> </tr> <tr> <td>Operation</td> <td>Select an Operation from the list of operations under the selected Pool Package.</td> </tr> <tr> <td colspan="2">Link Event Type</td> </tr> <tr> <td>Link Event</td> <td>Select a <i>Link Event</i> element (to act as a target for this <i>Link Event</i>) from the list of <i>Link Event</i> elements under the current <i>Process / Sub-Process</i>.</td> </tr> </tbody> </table>	Field	Usage	Message Event Type		Pool Package	Select the package that represents one of involved <i>Participants</i> .	Operation	Select an Operation from the list of operations under the selected Pool Package .	Link Event Type		Link Event	Select a <i>Link Event</i> element (to act as a target for this <i>Link Event</i>) from the list of <i>Link Event</i> elements under the current <i>Process / Sub-Process</i> .	
Field	Usage													
Message Event Type														
Pool Package	Select the package that represents one of involved <i>Participants</i> .													
Operation	Select an Operation from the list of operations under the selected Pool Package .													
Link Event Type														
Link Event	Select a <i>Link Event</i> element (to act as a target for this <i>Link Event</i>) from the list of <i>Link Event</i> elements under the current <i>Process / Sub-Process</i> .													

	<table border="1"> <tr> <td colspan="2">Timer Event Type</td> </tr> <tr> <td>Time Cycle</td> <td>Specify the value for the time duration.</td> </tr> <tr> <td>Time Date</td> <td>Specify the value for the time date.</td> </tr> <tr> <td colspan="2"> </td> </tr> <tr> <td colspan="2">Compensation Event Type</td> </tr> <tr> <td>Activity</td> <td>Select an <i>Activity</i> from the list of Activity elements under the current <i>Process / Sub-Process</i>.</td> </tr> <tr> <td colspan="2"> </td> </tr> <tr> <td colspan="2">Error Event Type</td> </tr> <tr> <td>Error</td> <td>Select an <i>Error</i> element.</td> </tr> <tr> <td colspan="2"> </td> </tr> </table>	Timer Event Type		Time Cycle	Specify the value for the time duration.	Time Date	Specify the value for the time date.			Compensation Event Type		Activity	Select an <i>Activity</i> from the list of Activity elements under the current <i>Process / Sub-Process</i> .			Error Event Type		Error	Select an <i>Error</i> element.			
Timer Event Type																						
Time Cycle	Specify the value for the time duration.																					
Time Date	Specify the value for the time date.																					
Compensation Event Type																						
Activity	Select an <i>Activity</i> from the list of Activity elements under the current <i>Process / Sub-Process</i> .																					
Error Event Type																						
Error	Select an <i>Error</i> element.																					
Assignments	Select one or more <i>Assignment</i> elements created in the <i>SupportingElements</i> package in this tab (Optional).	Assignment ^[1613] SupportingElements ^[1598]																				
OK	Save the values entered in the dialog.																					
Cancel	Discard the values entered in the dialog.																					
Help	Display this Help topic.																					
General	Open the UML Properties dialog.	Properties ^[843]																				

Notes

- **Cancel, Conditional, Escalation, Multiple, None, Parallel Multiple** and **Signal** event type cannot be mapped to BPEL 2.0
- The Assignments tab is not available on *Intermediate Events* that are attached to the boundary of an *Activity*
- Either set **Time Cycle** or **Time Date** for **Timer Trigger Type** but not both as they are mutually exclusive fields
- The **Error** event type is valid only on an *Intermediate Event* attached to the boundary of an *Activity*
- The **Compensation** event type is valid only on an *Event Sub-Process* or when attached to the boundary of an *Activity*

- The **Link Intermediate Event** can be used either as a *GOTO* or an *off-page* connector; therefore, this event can have either incoming or outgoing *Sequence Flows* - but not both

8.3.5.2.2.3 Activity

A BPMN 2.0 *Activity* represents work that is performed within a *Process*. An *Activity* can be modeled as a:

- **Sub-Process** - a compound *Activity* that is defined as a flow of other BPMN 2.0 elements or
- **Task** - an atomic *Activity* that cannot be broken down into a smaller unit

Activities - both *Tasks* and *Sub-Processes* - can also, optionally, act as *Looping constructs*. The *OMG BPMN 2.0 Specification* defines two types of *Looping construct*:

- **Standard Loop** (*while* or *until*)
- **Multi-Instance Loop** (*for each*)

BPEL Properties:

- Double-click on the *Activity* in the BPEL diagram
- Right-click on the *Activity* | **BPEL | BPEL 2.0 Properties**

Reference

Field/Button	Usage	See also
Name	Specify the name for the <i>Activity</i> .	
Type	Specify whether the <i>Activity</i> is a: <ul style="list-style-type: none"> • Task or • Sub-Process 	
Task Type / SubProcess Type	Depending on the value selected in the Type field, the Task Type / SubProcess Type field has the following values: <p>Task Type :</p> <ul style="list-style-type: none"> • Abstract • Business Rule • Manual • Receive • Script • Send • Service • User 	

	<p>SubProcess Type :</p> <ul style="list-style-type: none"> • Ad-Hoc • Call Activity • Compensation • Embedded • Event • Transaction 									
<p>Details</p>	<p>Depending on the selected Task Type, the Details tab changes as follows:</p> <table border="1" data-bbox="496 763 1209 1144"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Receive / Send / Service Task Type</td> </tr> <tr> <td>Pool Package</td> <td>Select the package that represents one of involved <i>Participants</i>.</td> </tr> <tr> <td>Operation</td> <td>Select an Operation from the list of operations for the selected Pool Package.</td> </tr> </tbody> </table>	Field	Usage	Receive / Send / Service Task Type		Pool Package	Select the package that represents one of involved <i>Participants</i> .	Operation	Select an Operation from the list of operations for the selected Pool Package .	
Field	Usage									
Receive / Send / Service Task Type										
Pool Package	Select the package that represents one of involved <i>Participants</i> .									
Operation	Select an Operation from the list of operations for the selected Pool Package .									
<p>Assignments</p>	<p>Select one or more <i>Assignment</i> elements created in the <i>SupportingElements</i> package in this tab (Optional).</p>	<p>Assignment^[1613] SupportingElements^[1598]</p>								
<p>Loop Details</p>	<p>Activities can be repeated sequentially, behaving like a loop. Specify the <i>Activity</i> looping details in this tab (Optional).</p> <table border="1" data-bbox="496 1464 1209 1991"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Loop Type</td> <td>Select: <ul style="list-style-type: none"> • Standard for creating a <i>while</i> or <i>until</i> loop. • Multinstance for creating a <i>for each</i> loop. </td> </tr> <tr> <td>Evaluate condition at the beginning of loop</td> <td>Check this option to create a <i>while Standard</i> loop. Defaults to unselected for an <i>until</i> loop.</td> </tr> <tr> <td>Loop Condition</td> <td>Enter a boolean condition for a Standard loop or a numeric condition for a Multinstance loop.</td> </tr> </tbody> </table>	Field	Usage	Loop Type	Select: <ul style="list-style-type: none"> • Standard for creating a <i>while</i> or <i>until</i> loop. • Multinstance for creating a <i>for each</i> loop. 	Evaluate condition at the beginning of loop	Check this option to create a <i>while Standard</i> loop. Defaults to unselected for an <i>until</i> loop.	Loop Condition	Enter a boolean condition for a Standard loop or a numeric condition for a Multinstance loop.	
Field	Usage									
Loop Type	Select: <ul style="list-style-type: none"> • Standard for creating a <i>while</i> or <i>until</i> loop. • Multinstance for creating a <i>for each</i> loop. 									
Evaluate condition at the beginning of loop	Check this option to create a <i>while Standard</i> loop. Defaults to unselected for an <i>until</i> loop.									
Loop Condition	Enter a boolean condition for a Standard loop or a numeric condition for a Multinstance loop.									

	Loop Maximum	Specify the maximum value for the loop even for a Standard loop (Optional).	
OK	Save the values entered in the dialog.		
Cancel	Discard the values entered in the dialog.		
Help	Display this Help topic.		
General	Open the UML Properties dialog.		Properties ^[843]

Notes

- The Assignments tab is not applicable for *Sub-Process*
- The Loop Details tab is not applicable for *Event Sub-Process*
- **Business Rule**, **Manual**, **Script** and **User Task** types cannot be mapped to BPEL 2.0
- **Ad-Hoc**, **Call Activity** and **Transaction Sub-Process** types cannot be mapped to BPEL 2.0

8.3.5.2.2.4 Gateway

A BPMN 2.0 *Gateway* controls the way in which *Sequence Flows* converge and diverge within a *Process*. They provide a gating mechanism that either allows or blocks a *Sequence Flow*.

BPEL Properties:

- Double-click on the *Gateway* in the BPEL diagram
- Right-click on the *Gateway* | **BPEL** | **BPEL 2.0 Properties**

Reference

Field/Button	Usage	See also
Name	Specify the name for the <i>Gateway</i> .	
Gateway	Select the type for the <i>Gateway</i> , namely: <ul style="list-style-type: none"> • Complex • Event • Exclusive • Inclusive 	

	<ul style="list-style-type: none"> Parallel 							
Details	Depending on the selected Gateway , the Details tab changes as follows: <table border="1" data-bbox="513 477 1299 725"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Event Gateway</td> </tr> <tr> <td>Event Type</td> <td>Set to Exclusive to represent an Exclusive Event Based Gateway.</td> </tr> </tbody> </table>	Field	Usage	Event Gateway		Event Type	Set to Exclusive to represent an Exclusive Event Based Gateway.	
Field	Usage							
Event Gateway								
Event Type	Set to Exclusive to represent an Exclusive Event Based Gateway.							
OK	Save the values entered in the dialog.							
Cancel	Discard the values entered in the dialog.							
Help	Display this Help topic.							
General	Open the UML Properties dialog.	Properties 8431						

Notes

- The target of the outgoing *Sequence Flows* of this **Event Exclusive Gateway** must be either a:
 - Receive Task** or a
 - Message** or **Timer Intermediate Event**
- If a **Message Task** is one of the targets of the outgoing *Sequence Flow* of an **Event Exclusive Gateway**, then a **Message Intermediate Event** cannot be the target of the other outgoing *Sequence Flows* of this **Event Exclusive Gateway**

8.3.5.2.2.5 End Event

A BPMN 2.0 *End Event* indicates where a particular *Process* ends. Every *Process* in Enterprise Architect must end with an *End Event*.

BPEL Properties:

- Double-click on the *End Event* in the BPEL diagram
- Right-click on the *End Event* | **BPEL** | **BPEL 2.0 Properties**

Reference

Field/Button	Usage	See also																
Name	Specify the name for the <i>End Event</i> .																	
Event Type	Select the type for the <i>End Event</i> , namely: <ul style="list-style-type: none"> • Cancel • Compensation • Error • Escalation • Message • Multiple • None • Signal • Terminate 																	
Details	<p>Depending on the selected Event Type, the Details tab changes as follows:</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Message Event Type</td> </tr> <tr> <td>Pool Package</td> <td>Select the package that represents one of the involved <i>Participants</i>.</td> </tr> <tr> <td>Operation</td> <td>Select an Operation from the list of operation the selected Pool Package.</td> </tr> <tr> <td colspan="2">Compensation Event Type</td> </tr> <tr> <td>Activity</td> <td>Select an <i>Activity</i> from the list of Activity element under the current <i>Process / Sub-Process</i>.</td> </tr> <tr> <td colspan="2">Error Event Type</td> </tr> <tr> <td>Error</td> <td>Select an <i>Error</i> element.</td> </tr> </tbody> </table>	Field	Usage	Message Event Type		Pool Package	Select the package that represents one of the involved <i>Participants</i> .	Operation	Select an Operation from the list of operation the selected Pool Package .	Compensation Event Type		Activity	Select an <i>Activity</i> from the list of Activity element under the current <i>Process / Sub-Process</i> .	Error Event Type		Error	Select an <i>Error</i> element.	
Field	Usage																	
Message Event Type																		
Pool Package	Select the package that represents one of the involved <i>Participants</i> .																	
Operation	Select an Operation from the list of operation the selected Pool Package .																	
Compensation Event Type																		
Activity	Select an <i>Activity</i> from the list of Activity element under the current <i>Process / Sub-Process</i> .																	
Error Event Type																		
Error	Select an <i>Error</i> element.																	

OK	Save the values entered in the dialog.	
Cancel	Discard the values entered in the dialog.	
Help	Display this Help topic.	
General	Open the UML Properties dialog.	Properties ^[843]

Notes

- The **Compensation** Event type is valid only on an *Event Sub-Process*
- **Cancel**, **Escalation**, **Multiple** and **Signal** Event types cannot be mapped to BPEL 2.0

8.3.5.2.2.6 *Data Object*

A BPMN 2.0 *Data Object* is mapped to a BPEL 2.0 *Variable* and occurs in the context of a Process or Sub-Process, just like a [Property](#) ^[1617]. A *Data Object* cannot have any incoming or outgoing *Sequence Flow* or *Message Flow*.

BPEL Properties:

- Double-click on the *Data Object* in the BPEL diagram
- Right-click on the *Data Object* | **BPEL | BPEL 2.0 Properties**

Reference

Field/Button	Usage	See also
Name	Specify the name for the <i>Data Object</i> .	
Type	Either: <ul style="list-style-type: none"> • Select an XML Schema <i>basic</i> type from the drop-down list or • Use the (...) button to select XML Schema <i>ComplexType</i>, <i>SimpleType</i>, <i>Union</i> or <i>Enumeration</i> 	
OK	Save the values entered in the dialog.	
Cancel	Discard the values entered in the dialog.	

Help	Display this Help topic.	
General	Open the UML Properties dialog.	Properties [843]

8.3.5.2.2.7 Property

A BPMN 2.0 *Property* is mapped to a BPEL 2.0 *Variable* and occurs in the context of a *Process* or *Sub-Process*, just like a [Data Object](#)^[1610].

However, unlike a *Data Object*, *Property* is not displayed on a BPEL diagram. Enterprise Architect supports *Property* on *BPMN 2.0 BPEL* and *BPMN 2.0 Activity* elements.

Access [View](#) | [Diagram Toolbox: More tools](#) | [BPMN 2.0](#) | [BPMN 2.0 Business Process](#) | [BPMN 2.0 Types](#)

BPEL Properties:

- Right-click on the *Property* | [BPEL](#) | [BPEL 2.0 Properties](#)

Reference

Field/Button	Usage	See also
Name	Specify the name for the <i>Property</i> .	
Type	Either: <ul style="list-style-type: none"> • Select an XML Schema <i>basic</i> type from the drop-down list or • Use the (...) button to select XML Schema <i>ComplexType</i>, <i>SimpleType</i>, <i>Union</i> or <i>Enumeration</i> 	
OK	Save the values entered in the dialog.	
Cancel	Discard the values entered in the dialog.	
Help	Display this Help topic.	
General	Open the UML Properties dialog.	Properties [887]

8.3.5.2.2.8 Sequence Flow

A BPMN 2.0 *Sequence Flow* connector shows the order in which the activities (*Tasks* and *Events*) are performed in a *BPEL Process*.

BPEL Properties:

- Double-click on the *Sequence Flow* in the BPEL diagram
- Right-click on the *Sequence Flow* | **BPEL** | **BPEL 2.0 Properties**

Reference

Field/Button	Usage	See also
Condition Type	Specify the type of the condition on the <i>Sequence Flow</i> , namely: <ul style="list-style-type: none"> • None • Default • Expression 	
Expression	This field is enabled when the Condition Type is set to Expression . Specify a boolean expression to act as a gating condition.	
Ordering	This field is enabled when the Condition Type is set to Expression . Specify a numerical value that determines the order in which the condition set in the Expression field is to be evaluated.	
OK	Save the values entered in the dialog.	
Cancel	Discard the values entered in the dialog.	
Help	Display this Help topic.	
General	Open the UML Properties dialog.	Properties 843

8.3.5.2.2.9 Pool

A BPMN 2.0 *Pool* represents a Participant in a *Process* and does not map to any specific BPEL 2.0 element. Enterprise Architect uses *Pools* to represent external Participants, with which the *BPEL Process* communicates. These are 'black box' *pools*; that is, they are abstract and do not expose any details (they do not contain any BPMN 2.0 elements inside them).

BPEL Properties:

- Double-click on the *Pool* in the BPEL diagram
- Right-click on the *Pool* | **BPEL | BPEL 2.0 Properties**

Reference

Field/Button	Usage	See also
Name	Specify the name for the <i>Pool</i> .	
OK	Save the values entered in the dialog.	
Cancel	Discard the values entered in the dialog.	
Help	Display this Help topic.	
General	Open the UML Properties dialog.	Properties [843]

Notes

- A *BPEL Process* should not contain a *Pool* as its child element
- A BPEL diagram under a *BPEL Process* contains an implicit *Pool*, so it is invalid to add a *Pool* in this BPEL diagram to represent the *BPEL Process*
- *Pool* cannot have any incoming or outgoing *Sequence Flow* connectors - it can have only incoming or outgoing *Message Flow* connectors

8.3.5.2.2.10 Assignment

A BPMN 2.0 *Assignment* element enables data to be copied between messages and variables within a *BPEL Process*. An *Assignment* element maps to a BPEL 2.0 *assign* activity and copies the specified value from the source to the target.

In Enterprise Architect, *Assignment* elements should be created in the *Assignments* package in *SupportingElements*. If they are created elsewhere, they cannot be enacted correctly.


Access [View | Diagram Toolbox: More tools | BPMN 2.0 | BPMN 2.0 Business Process | BPMN 2.0 Types](#)

BPEL Properties:

- Double-click on the *Assignment* in the **BPEL** diagram
- Right-click on the *Assignment* | **BPEL | BPEL 2.0 Properties**

Reference

Field/Button	Usage	See also
Name	Specify the name for the <i>Assignment</i> .	
Copy From		
Type	Depending on the value selected in this field, further details are required.	
Literal	Specify a literal value.	
Expression	Specify an expression.	
Message	Select a BPMN 2.0 <i>Message</i> in the package representing the <i>BPEL Process / Pool</i> under the <i>SupportingElements</i> package.	SupportingElement s ^[1598] Web Service Operation ^[1615]
Part	Select a BPMN 2.0 <i>Property</i> belonging to the selected Message .	
Variable	Select a BPMN 2.0 <i>Data Object</i> or BPMN 2.0 <i>Property</i> created under the <i>BPEL Process</i> .	
Copy To		
Message	Select a BPMN 2.0 <i>Message</i> in the package representing the <i>BPEL Process / Pool</i> under the <i>SupportingElements</i> package.	SupportingElement s ^[1598] Web Service Operation ^[1615]
Part	Select a BPMN 2.0 <i>Property</i> belonging to the selected Message .	
Variable	Select a BPMN 2.0 <i>Data Object</i> or BPMN 2.0 <i>Property</i> created under the <i>BPEL Process</i> .	
OK	Save the values entered in the dialog.	
Cancel	Discard the values entered in the dialog.	

Help	Display this Help topic.	
General	Open the UML Properties dialog.	Properties 

Notes



- *Messages* are created when you create a Web Service Operation

8.3.5.2.3 Create BPEL 2.0 Web Service Operation


BPEL is an **Orchestration** language, which orchestrates services that are exposed using **WSDL 1.1**. It coordinates the execution of the various operations of these involved services. BPEL 2.0 supports **WSDL 1.1 one-way** and **request-response** operations only (and not **solicit-response** and **notification** operations). Enterprise Architect enables you to create, for *BPEL Process* and *Pool* elements, **Web Service Operations** that support either *synchronous* (request-response) or *asynchronous* (one-way) interactions.

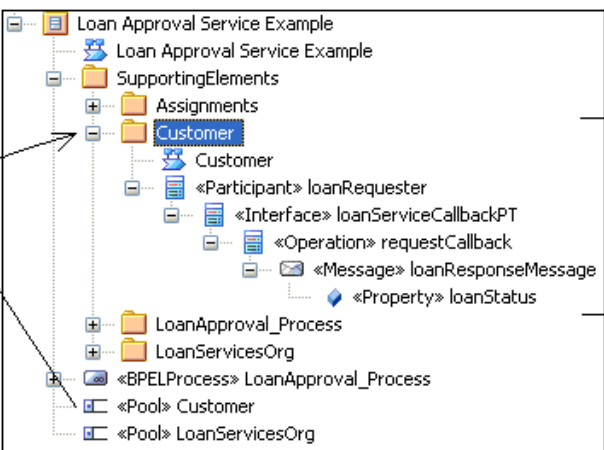
Access Right-click on a BPEL Process or Pool | BPEL | Create WebService

Reference

Field/Button	Usage	See also						
Operation	Specify whether to create a new Web Service Operation or create one from an existing WSDL created / imported into Enterprise Architect (using the Import WSDL dialog) by selecting either: <ul style="list-style-type: none"> • Create New or • Create from existing WSDL PortType Operation 	Model WSDL  Import WSDL 						
Type	Specify whether to create a <i>one-way</i> or <i>request-response</i> Operation by selecting either : <ul style="list-style-type: none"> • Asynchronous or • Synchronous 							
Partnership Details	The interaction between the Web Service and the <i>BPEL Process</i> is modeled as a BPEL <i>partnerLink</i> . For this interaction, provide: <table border="1" data-bbox="491 1780 1220 2038"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Name of the BPEL <i>partnerLink</i>.</td> </tr> <tr> <td>Role</td> <td>Name of either:</td> </tr> </tbody> </table>	Field	Usage	Name	Name of the BPEL <i>partnerLink</i> .	Role	Name of either:	
Field	Usage							
Name	Name of the BPEL <i>partnerLink</i> .							
Role	Name of either:							

		<ul style="list-style-type: none"> • <i>myRole</i> (if this operation belongs to <i>BPEL Process</i> Web Service) or • <i>partnerRole</i> (if this operation belongs to Web Service of a <i>Pool</i>) <p>of the BPEL <i>partnerLink</i>.</p>																			
<p>Web Service</p>	<table border="1"> <thead> <tr> <th data-bbox="483 622 735 696">Field</th> <th data-bbox="735 622 1222 696">Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2" data-bbox="483 696 1222 752">When Create New is selected in the Operation field:</td> </tr> <tr> <td data-bbox="483 752 735 853">PortType</td> <td data-bbox="735 752 1222 853">Name of the WSDL 1.1 PortType (Interface</td> </tr> <tr> <td data-bbox="483 853 735 954">Operation</td> <td data-bbox="735 853 1222 954">Name of the WSDL 1.1 PortType Operation.</td> </tr> <tr> <td colspan="2" data-bbox="483 954 1222 1010"></td> </tr> <tr> <td colspan="2" data-bbox="483 1010 1222 1099">When Create from existing WSDL PortType Operation is selected the Operation field:</td> </tr> <tr> <td data-bbox="483 1099 735 1223">WSDL Package</td> <td data-bbox="735 1099 1222 1223">Select an existing WSDL package created / imported into Enterprise Architect.</td> </tr> <tr> <td data-bbox="483 1223 735 1357">PortType</td> <td data-bbox="735 1223 1222 1357">Select a WSDL 1.1 PortType from the selected WSDL Package.</td> </tr> <tr> <td data-bbox="483 1357 735 1491">Operation</td> <td data-bbox="735 1357 1222 1491">Select a WSDL 1.1 PortType Operation in the selected PortType.</td> </tr> </tbody> </table>	Field	Usage	When Create New is selected in the Operation field:		PortType	Name of the WSDL 1.1 PortType (Interface	Operation	Name of the WSDL 1.1 PortType Operation .			When Create from existing WSDL PortType Operation is selected the Operation field:		WSDL Package	Select an existing WSDL package created / imported into Enterprise Architect.	PortType	Select a WSDL 1.1 PortType from the selected WSDL Package .	Operation	Select a WSDL 1.1 PortType Operation in the selected PortType .		
Field	Usage																				
When Create New is selected in the Operation field:																					
PortType	Name of the WSDL 1.1 PortType (Interface																				
Operation	Name of the WSDL 1.1 PortType Operation .																				
When Create from existing WSDL PortType Operation is selected the Operation field:																					
WSDL Package	Select an existing WSDL package created / imported into Enterprise Architect.																				
PortType	Select a WSDL 1.1 PortType from the selected WSDL Package .																				
Operation	Select a WSDL 1.1 PortType Operation in the selected PortType .																				
<p>Input</p>	<table border="1"> <thead> <tr> <th data-bbox="483 1547 735 1621">Field</th> <th data-bbox="735 1547 1222 1621">Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2" data-bbox="483 1621 1222 1677">When Create New is selected in the Operation field:</td> </tr> <tr> <td data-bbox="483 1677 735 1778">Message Name</td> <td data-bbox="735 1677 1222 1778">Name of the WSDL 1.1 Message.</td> </tr> <tr> <td data-bbox="483 1778 735 1901">Properties</td> <td data-bbox="735 1778 1222 1901">Click on the (...) button to enter the WSDL 1.1 Message Part Name and XSD Type.</td> </tr> <tr> <td colspan="2" data-bbox="483 1901 1222 1957"></td> </tr> <tr> <td colspan="2" data-bbox="483 1957 1222 2042">When Create from existing WSDL PortType Operation is selected</td> </tr> </tbody> </table>	Field	Usage	When Create New is selected in the Operation field:		Message Name	Name of the WSDL 1.1 Message .	Properties	Click on the (...) button to enter the WSDL 1.1 Message Part Name and XSD Type .			When Create from existing WSDL PortType Operation is selected									
Field	Usage																				
When Create New is selected in the Operation field:																					
Message Name	Name of the WSDL 1.1 Message .																				
Properties	Click on the (...) button to enter the WSDL 1.1 Message Part Name and XSD Type .																				
When Create from existing WSDL PortType Operation is selected																					

	<p>the Operation field:</p> <ul style="list-style-type: none"> The fields in this tab are pre-filled with the details of the input WSDL 1.1 Message (of the <i>WSDL 1.1 PortType Operation</i> selected in the Operation field in the Web Service tab) 															
Output	<table border="1"> <thead> <tr> <th data-bbox="493 560 737 618">Field</th> <th data-bbox="737 560 1216 618">Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2" data-bbox="493 618 1216 676">When Create New is selected in the Operation field:</td> </tr> <tr> <td data-bbox="493 676 737 779">Message Name</td> <td data-bbox="737 676 1216 779">Name of the WSDL 1.1 Message.</td> </tr> <tr> <td data-bbox="493 779 737 904">Properties</td> <td data-bbox="737 779 1216 904">Click on the (...) button to enter the WSDL 1.1 Message Part Name and XSD Type.</td> </tr> <tr> <td colspan="2" data-bbox="493 904 1216 963"></td> </tr> <tr> <td colspan="2" data-bbox="493 963 1216 1021">When Create from existing WSDL PortType Operation is selected in the Operation field:</td> </tr> <tr> <td colspan="2" data-bbox="493 1021 1216 1227"> <ul style="list-style-type: none"> The fields in this tab are pre-filled with the details of the output WSDL 1.1 Message (of the <i>WSDL 1.1 PortType Operation</i> selected in the Operation field in the Web Service tab) </td> </tr> </tbody> </table>	Field	Usage	When Create New is selected in the Operation field:		Message Name	Name of the WSDL 1.1 Message .	Properties	Click on the (...) button to enter the WSDL 1.1 Message Part Name and XSD Type .			When Create from existing WSDL PortType Operation is selected in the Operation field:		<ul style="list-style-type: none"> The fields in this tab are pre-filled with the details of the output WSDL 1.1 Message (of the <i>WSDL 1.1 PortType Operation</i> selected in the Operation field in the Web Service tab) 		
Field	Usage															
When Create New is selected in the Operation field:																
Message Name	Name of the WSDL 1.1 Message .															
Properties	Click on the (...) button to enter the WSDL 1.1 Message Part Name and XSD Type .															
When Create from existing WSDL PortType Operation is selected in the Operation field:																
<ul style="list-style-type: none"> The fields in this tab are pre-filled with the details of the output WSDL 1.1 Message (of the <i>WSDL 1.1 PortType Operation</i> selected in the Operation field in the Web Service tab) 																
OK	<p>Create a Web Service Operation based on the values entered in the dialog.</p> <p>The operation is created in a package that has the same name as the <i>BPEL Process / Pool</i> from which this dialog is invoked, under the <i>SupportingElements</i> package.</p>	SupportingElements 														

	 <p>A Package will be created under the <i>SupportingElements</i> package, with the same name as that of the <i>Pool</i>. This Package will act as a container for all the Web Service Operations created for that <i>Pool</i>.</p>	
<p>Cancel</p>	<p>Discard the values entered in the dialog and abort creating the Web Service Operation.</p>	
<p>Help</p>	<p>Display this Help topic.</p>	

Notes

- The Output tab is not applicable for an Asynchronous operation

8.3.5.2.4 Generate BPEL 2.0

BPEL 2.0 code can be generated from a *BPEL Process*. Enterprise Architect validates the *BPEL Process* before generating the BPEL 2.0 code. In addition to generating the BPEL 2.0 code, WSDL 1.1 files are generated for the *BPEL Process* and all the involved *Pools* (provided Web Service Operations are defined for them).

Access Right-click on BPEL Process | BPEL | Generate BPEL 2.0

Reference

Field/Button	Usage	See also
<p>File Name</p>	<p>Specify the path where the BPEL 2.0 file is to be generated.</p>	
<p>Namespace Details</p>	<p>Double-click on an entry (if any) in this field to open the Namespace Details dialog and add / edit the namespace details.</p> <p>The entry DefaultPool represents the current <i>BPEL Process</i>.</p>	

Generate BPEL	Validate the model and generate BPEL 2.0.	BPEL Model Validation ^[1619]
Close	Close this dialog.	
Help	Display this Help topic.	
View BPEL	View the generated BPEL 2.0 file.	

Learn more

- [Model a BPEL 2.0 Process](#) ^[1599]
- [Create a Web Service Operation](#) ^[1615]

Learning Center topics

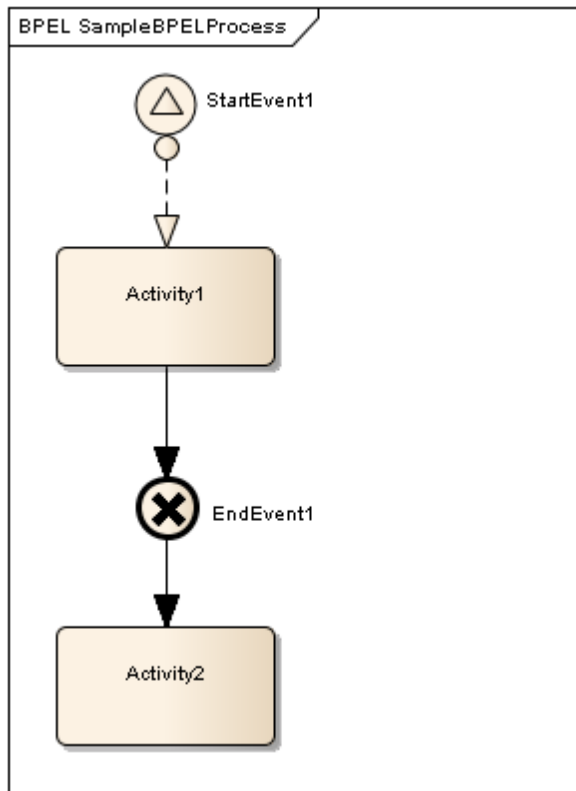
- **(Alt + F1) | Business Modeling | BPEL | Generate BPEL**

8.3.5.3 BPEL Model Validation

The BPEL model is validated for both *syntax* and *semantic* errors before generating the BPEL code. Upon successful validation, the BPEL file (and the associated WSDL 1.1 files, if any) are generated. The results of the validation, as well as the progress of the BPEL generation, are displayed in the BPEL Progress tab of the Output window.

Example BPEL Model Violation:

The following model shows several basic BPEL violations:



- *StartEvent1* is of type **Signal**, which cannot be mapped to BPEL
- The *Message Flow* connector between *StartEvent1* and *Activity1* is invalid, as *Message Flow* cannot be used to connect *Activities* or *Events* within a *Process*
- *EndEvent1* is of type **Cancel**, which cannot be mapped to BPEL
- *EndEvent1* cannot have any outgoing *Sequence Flow* connector, as it represents the end of a *Process*
- *Activity2* is not valid, as only an *End Event* represents the end of a *Process*

Notes

- Double-click on a validation error entry in the BPEL Progress tab of the Output window to go to the source of the error in the Project Browser

Learn more

- [Generate BPEL 1.1](#)^[1594]
- [BPEL 1.1 Modeling Restrictions](#)^[1571]
- [Generate BPEL 2.0](#)^[1618]
- [BPEL 2.0 Modeling Restrictions](#)^[1595]
- [System Output Window](#)^[164]

Learning Center topics

- (Alt + F1) | [Business Modeling](#) | [Business Rules](#) | [Validate Business Rules](#)

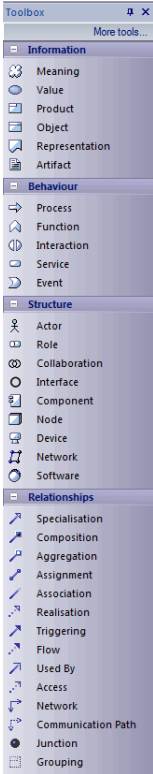
8.3.6 ArchiMate

ArchiMate is an open-standard enterprise architecture language from The Open Group, based on the IEEE 1471 standard. It offers a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems and technical infrastructure, enabling Enterprise Architects to describe, analyse and visualize the relationships among business domains in an unambiguous way.

Access [View](#) | [Diagram Toolbox: More Tools](#) | [ArchiMate](#)

Topics

Images	Topic	Detail	See also
	ArchiMate in Enterprise Architect	<p>Enterprise Architect enables you to develop ArchiMate diagrams quickly and simply, through use of an ArchiMate MDG Technology integrated with the Enterprise Architect installer. The ArchiMate facilities are provided in the form of:</p> <ul style="list-style-type: none"> • An ArchiMate diagram type, accessed through the New Diagram dialog • A set of ArchiMate pages in the Toolbox • ArchiMate element and relationship entries in the Toolbox Shortcut Menu and Quick Linker 	<p>New Diagram [724]</p> <p>Toolbox Shortcut [704]</p> <p>Quick Linker [791]</p>

Images	Topic	Detail	See also
	<p>ArchiMate Toolbox Pages</p>	<p>The toolbox pages provide three categories of elements - Information, Behavior and Structure - and a page of connectors that are largely based on the UML connectors.</p> <p>The appearance of elements can be modified with the use of Tagged Values, as suggested below:</p> <ul style="list-style-type: none"> For <i>Artifact, Process, Function, Interaction, Service, Event, Actor, Role, Collaboration, Interface, Component, Node</i> and <i>Device</i> elements: <code>iconstyle=true</code> shows the iconic representation of the element <code>iconstyle=false</code> shows a rectangle with a decoration in the top corner An Interface element (if <code>iconstyle=true</code>) can be provided, required, symmetric or assembly, and can be rotated, by setting the Tagged Values A Function element (if <code>iconstyle=true</code>) can be rotated An Actor element (if <code>iconstyle=false</code>) can be compound Process and Function elements can be marked <i>atomic</i> <p>Because there is a large range of ArchiMate elements, you should make good use of the Quick Linker arrow to guide you in selecting appropriate source and target elements and relationship types to model your enterprise architecture.</p>	
	<p>Disable ArchiMate</p>	<p>If you prefer not to use ArchiMate in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).</p>	<p>MDG Technologies ¹²⁸⁵</p>

Learning Center topics

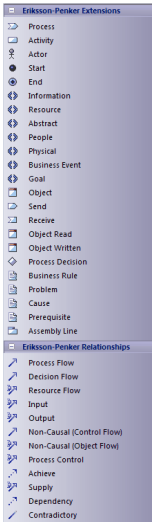
- **(Alt + F1) | Modeling Languages | Business | ArchiMate**

8.3.7 Eriksson-Penker Extensions

Eriksson-Penker extensions (developed by H. E. Eriksson and M.Penker) provide a framework for UML business processing model extensions, to which an enterprise architect can add stereotypes and properties appropriate to their business.

Access [View | Diagram Toolbox: More Tools | Eriksson-Penker Extensions](#)

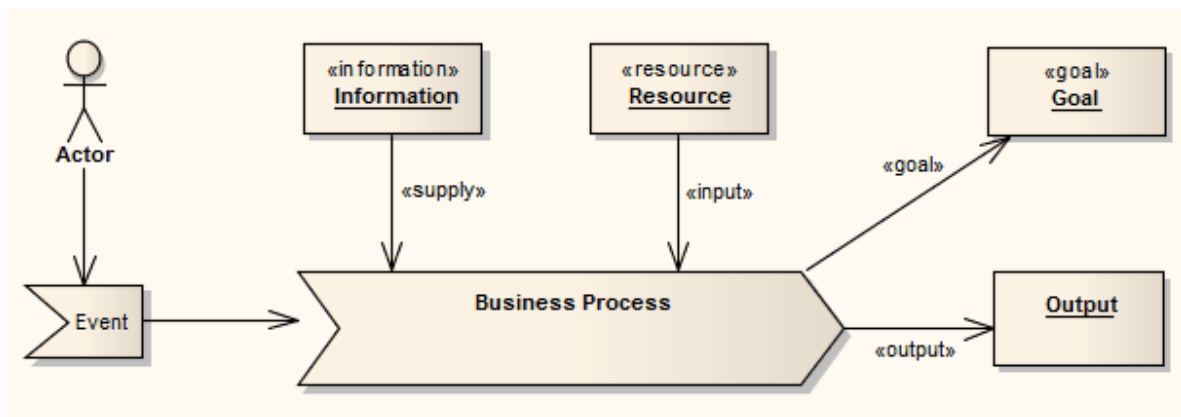
Topics

Images	Topic	Detail	See also
	Eriksson-Penker in Enterprise Architect	<p>Enterprise Architect provides - through the integration of MDG Technologies with the installer - two well-respected and proven UML extensions that further enhance the capture of business activities, processes, objects and information flows.</p> <p>One of these is Business Process Modeling Notation (BPMN).</p> <p>The other is the Eriksson-Penker profile which, through a set of stereotypes, provides a unique and powerful means of visualizing and communicating business processes and the necessary flow of information within an organization.</p> <p>The Eriksson-Penker extensions are provided in the form of:</p> <ul style="list-style-type: none"> • An Eriksson-Penker diagram type, accessed through the New Diagram dialog • An Eriksson-Penker page in the Toolbox • Eriksson-Penker element and relationship entries in the Toolbox Shortcut Menu and Quick Linker 	<p>Business Process Modeling Notation^[1545]</p> <p>New Diagram^[724]</p> <p>Toolbox Shortcut^[704]</p> <p>Quick Linker^[791]</p>
	Eriksson-Penker Toolbox Page	See the image to the left .	

Images	Topic	Detail	See also

Example

The following is an example of a simple Eriksson-Penker diagram:



Learn more

- [The Business Process Model](#) (Online Resource)

Learning Center topics

- **(Alt + F1) | Modeling Languages | Business | Eriksson Penker**

8.4 Data Models

Data Modelers and Information Architects are challenged with creating data models that span multiple levels of abstraction - from concept to physical implementation. They might also be responsible for maintaining traceability between these models. Enterprise Architect helps to meet these challenges with easy-to-use tools for building and maintaining all of the fundamental data models - Conceptual, Logical and Physical.

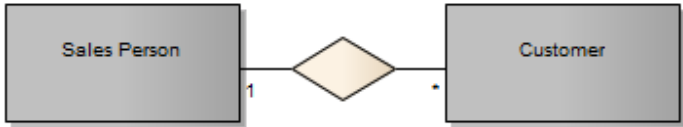

Topics

Topic	Detail	See also
Conceptual Data Models	<p>Conceptual data models, also called Domain models, establish the basic concepts and semantics of a given domain and help to communicate these to a wide audience of stakeholders.</p> <p>Conceptual models also serve as a common vocabulary during the analysis stages of a project; they can be created in Enterprise Architect using Entity-Relationship or UML Class models.</p>	Conceptual Data Model ^[1626]
Logical Data Models	<p>Logical data models add further detail to conceptual model elements and refine the structure of the domain; they can be defined using Entity-Relationship or UML Class models.</p> <p>One benefit of a Logical data model is that it provides a foundation on which to base the Physical model and subsequent database implementation.</p>	Logical Data Model ^[1627] Entity Relationship Diagrams (ERDs) ^[1627]
Logical to Physical Model	A Logical model can be transformed to a Physical Data model using a DDL Transformation.	DDL Transformation ^[1689]
Physical Data Models	<p>Physical data models in Enterprise Architect help you visualize your database structure and automatically derive the corresponding database schema; you use Enterprise Architect's UML Profile for Data Modeling specifically for this purpose.</p> <p>The profile provides useful extensions of the UML standard that map database concepts of tables and relationships onto the UML concepts of Classes and Associations; you can also model database keys, triggers, constraints, referential integrity and other relational database features.</p> <p>Because Enterprise Architect lets you visualize each type of data model in the same repository, you can easily manage dependencies between each level of abstraction; this helps you maximize traceability and verify completeness of system implementation.</p>	Database Engineering ^[1931] Physical Data Model ^[1932]

8.4.1 Conceptual Data Model

A Conceptual data model is the most abstract form of data model. It is helpful for communicating ideas to a wide range of stakeholders because of its simplicity. Therefore platform-specific information, such as data types, is omitted from a Conceptual data model. Other implementation details, such as procedures and interface definitions, are also excluded. Below is an example of a Conceptual data model that is rendered using two of the notations supported by Enterprise Architect.

Topics

Topic	Detail	See also
<p>Example</p>	<p>Using Entity-Relationship (ER) notation, we represent the data concepts <i>Sales Person</i> and <i>Customer</i> as Entities with a 1-to-many relationship between them.</p>  <p>A Conceptual data model that uses Entity-Relationship notation</p> <p>We can represent the exactly the same semantic information using UML Classes and Associations.</p>  <p>An equivalent model using UML Class notation</p> <p>Whether you use UML or ER notation to represent data concepts in your project depends on the experience and preferences of the stakeholders involved.</p> <p>The detailed structure of the data concepts illustrated above is defined later by the Logical data model.</p>	<p>Entity Relationship Diagrams (ERDs) ^[1627]</p> <p>UML Class Modeling ^[1014]</p>

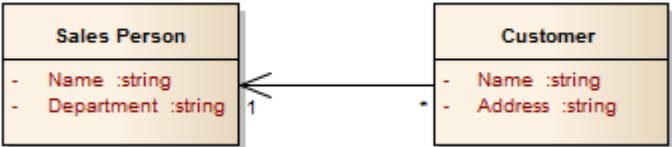
Learn more

- [Logical data models](#) ^[1627]
- [Entity Relationship modeling](#) ^[1627]
- [UML Class modeling](#) ^[1014]

8.4.2 Logical Data Model

Logical data models help to define the detailed structure of the data elements in a system and the relationships between data elements. They refine the data elements introduced by a Conceptual data model and form the basis of the Physical data model. In Enterprise Architect, a Logical data model is typically represented using the UML Class notation.

Topics

Topic	Detail	See also
Example	<p>A simple example of a Logical data model is provided below.</p> <p>Note that the data elements <i>Sales Person</i> and <i>Customer</i> contain UML Attributes; the Attribute types, however, remain platform-independent.</p>  <pre> classDiagram class SalesPerson { - Name :string - Department :string } class Customer { - Name :string - Address :string } SalesPerson "1" -- "*" Customer </pre> <p>A Logical data model defined using UML Class notation</p> <p>Platform-specific Attribute types and other meta-data that relate to a specific DBMS implementation are defined by the Physical data model.</p>	<p>UML Class Modeling ^[1014]</p>

Learn more

- [Physical data models](#) ^[1932]
- [Connect to Element Feature](#) ^[944]

8.4.3 Entity Relationship Diagrams (ERDs)

According to the online [Wikipedia](#):

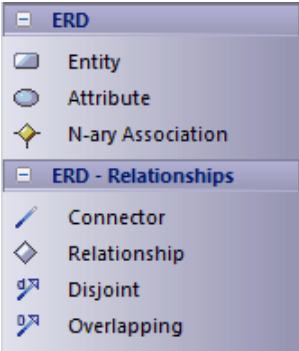
An entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called Entity-Relationship Diagrams, ER Diagrams, or ERDs.

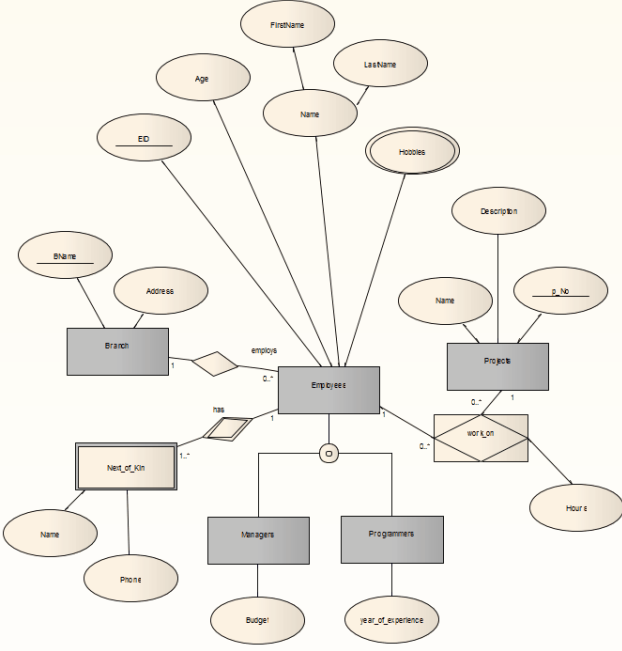
Entity Relationship Diagrams in Enterprise Architect:

Entity Relationship Diagrams in Enterprise Architect are based on Chen's ERD building blocks: entities are represented as rectangles, attributes are represented as ellipses and relationships are represented as diamond-shape connectors.

ERD technology in Enterprise Architect assists you in every stage from building conceptual data models to generating Data Definition Language (DDL) for the target DBMS.

Topics

Topic	Detail	See also
<p>MDG Technology for ERD and ERD Transformations</p>	<p>Enterprise Architect enables you to develop Entity Relationship diagrams quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer.</p> <p>The Entity Relationship diagram facilities are provided in the form of:</p> <ul style="list-style-type: none"> • An Entity Relationship diagram type, accessed through the New Diagram dialog • An Entity Relationship Diagram page in the Toolbox • Entity Relationship element and relationship entries in the Toolbox Shortcut Menu and Quick Linker <p>Enterprise Architect also provides transformation templates to transform Entity Relationship Diagrams into Data Modeling Diagrams, and vice versa.</p>	<p>New Diagram Dialog^[724]</p> <p>Toolbox Shortcut^[704]</p> <p>Quick Linker^[794]</p> <p>ERD to Data Model Transformation^[1697]</p> <p>Data Model to ERD Transformation^[1689]</p>
<p>Entity Relationship Diagram Toolbox Page</p> 	<p>You can access the Entity Relationship Diagram page of the Toolbox through the More tools Entity Relationship Diagrams option.</p> <ul style="list-style-type: none"> • <i>Entity</i> is an object or concept that is uniquely identifiable; the property of Multiplicity in the SourceRole and TargetRole definitions for the <i>Relationship</i> connector (below) can be used to define the cardinality of an Entity that participates in this relationship • <i>Attribute</i> is a property of an entity or a relationship type • <i>N-ary Association</i> represents unary (many-to-many recursive) or ternary relationships and can also be used to represent relationships that have attributes among the entities; the N-ary Association element should always be at the target end of a connector • <i>Connector</i> is a connector between an Entity and an Attribute, and between two Attributes • <i>Relationship</i> is a diamond-shape connector, representing the meaningful association among entities • <i>Disjoint</i> and <i>Overlapping</i> represent the 	

Topic	Detail	See also
	relationships between the super-class Entity and the sub-class Entity	
<p>A typical Entity Relationship diagram</p>		
<p>Disable Entity Relationship Diagrams</p>	<p>If you prefer not to use Entity Relationship Diagrams in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).</p>	<p>MDG Technologies [1285]</p>

Tagged Values:

Some of the Entity Relationship diagram components can be modified by Tagged Values, as indicated below:

Component	Tagged Value	Notes
Entity	isWeakEntity	If true , this entity is a weak entity.
Attribute	attributeType	<p>Four options:</p> <ul style="list-style-type: none"> • normal Attribute • primary key attribute • multi-valued Attribute • derived Attribute

Component	Tagged Value	Notes
	commonDataType	Defines the common data type for each attribute.
	dbmsDataType	Defines the customized DBMS data type for each attribute. You must define the customized type first through the Settings Database Datatypes menu option. Also, set the <i>commonDataType</i> tag to na to activate the <i>dbmsDataType</i> tag.
N-ary Association	isRecursive	If true , the N-ary Association represents the many-to-many recursive relationship. For one-to-many and one-to-one recursive relationships, we suggest using the normal Relationship connector. Sometimes you might want to limit the stretch of the diamond-shape Relationship connectors; simply pick a Relationship connector, right-click to display the context menu, and select the Bend Line at Cursor option.
Relationship	isWeak	If true , the Relationship is a weak relationship.
Disjoin Overlapping	Participation	Two options: partial and total .

Notes

- Entity Relationship Diagrams are supported in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

8.4.4 Table



Description

A Table is a stereotyped Class. It is drawn with a small table icon in the upper right corner. You typically use this element in Data Modeling diagrams.

A Table element has a special Properties dialog, with settings for database type and the ability to set column information and data-related operations such as triggers and indexes. When setting up a Table, make sure you set the default database type for that Table, otherwise you do not have any data types to choose from when creating columns.

Toolbox icon

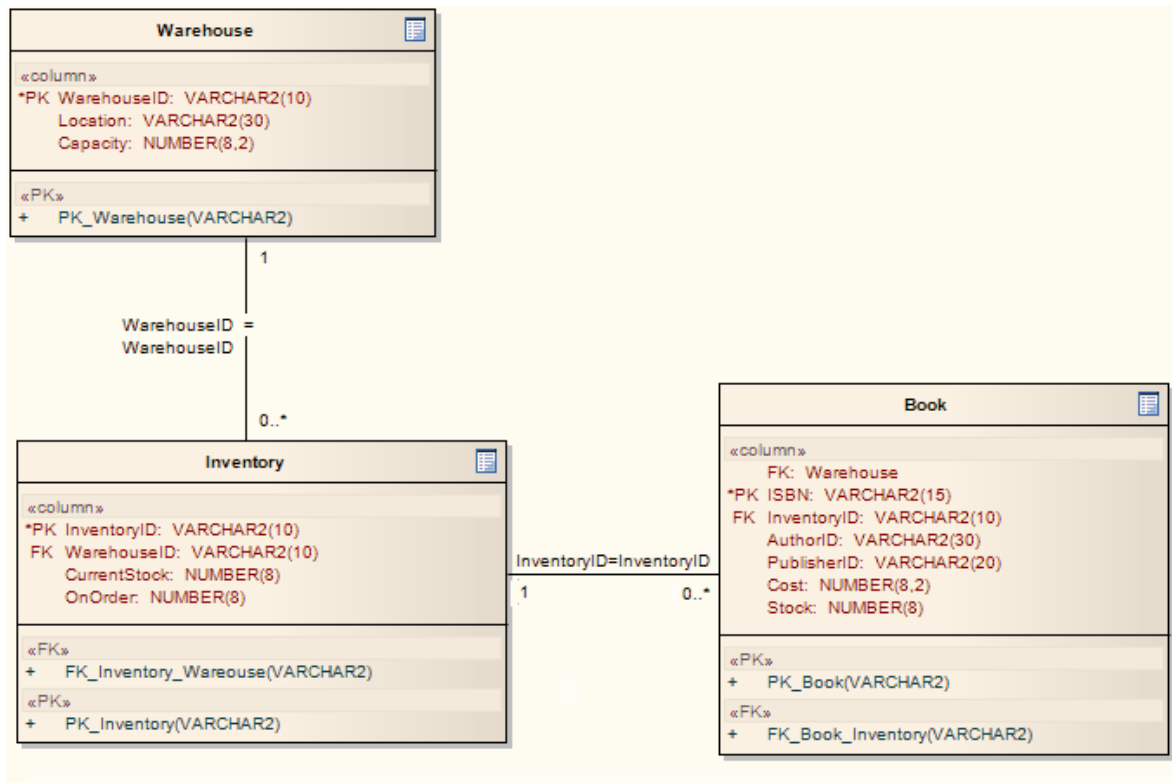


Learn more

- [Data Modeling Diagram](#)^[1631]
- [Set the Default Database Type](#)^[1937]

8.4.5 Database Schema

The following diagram shows an example Database Schema represented by a Physical Data Model.



Learn more

- [Physical Data Model](#)^[1932]
- [Data Modeling Toolbox](#)^[720]

8.5 Geodatabase Design for ArcGIS

Beta Implementation

This is a **BETA** release of the **ArcGIS** profile and associated tools. As such it is supplied without warranty of any kind and is subject to change without notice during this Beta period. Sparx Systems welcomes any feedback, issues, suggestions and comments on this implementation. As with all Beta software, please take due care when using the ArcGIS profile in a production environment.

ArcGIS, developed by **Esri**, is a system that supports the development and management of geodatabases. As for other databases, it is useful to model the design of a geodatabase using a standard notation such as UML. Enterprise Architect's UML profile for ArcGIS is part of the built-in MDG Technology for ArcGIS.

The MDG Technology for ArcGIS supports:

- Geodatabase design for the ArcGIS platform
- Import and export of ArcGIS workspaces

Notice of Acknowledgement:

Support for modeling ArcGIS databases in Enterprise Architect was developed in collaboration with the Commonwealth Scientific and Industrial Research Organization (**CSIRO**), who defined mappings between UML 2 and ArcGIS concepts, and prototyped an automated import and export capability for ArcGIS geodatabase schemas represented in UML.

Once you have modeled an ArcGIS schema in Enterprise Architect, you can **export the model to ArcGIS** as an XML Workspace document.

You can also visualize an existing ArcGIS geodatabase schema, by **importing the ArcGIS XML Workspace document** into Enterprise Architect.

Learn more

- [ArcGIS UML Profile](#)^[1633]
- [Export ArcGIS XML Workspace](#)^[1640]
- [Import ArcGIS XML Workspace](#)^[1641]

8.5.1 ArcGIS UML Profile

Enterprise Architect's UML profile for ArcGIS is part of the built-in MDG Technology for ArcGIS.

Access [Extensions | ArcGIS](#)

Topics

Topic	Detail	See also
Profile Support	The built-in MDG Technology for ArcGIS provides:	New Diagram ^[724]

Topic	Detail	See also
	<ul style="list-style-type: none"> • ArcGIS Toolbox pages that map ArcGIS concepts to appropriately stereotyped UML elements • A Model Pattern that helps you to start designing geodatabases quickly and to use the required package structure in Enterprise Architect • Datatypes that are specific to the ArcGIS platform • Quick-linker capabilities that help you make valid connections between elements 	Toolbox Shortcut [704]
		Quick Linker [79]
ArcGIS Toolbox Pages	<p>The ArcGIS Toolbox contains three pages:</p> <ul style="list-style-type: none"> • Core - for elements • Relationships - for connectors • Supported Types - for Attributes and elements 	ArcGIS Toolbox Pages [1634]
		Diagram Toolbox [698]
Spatial References	Enterprise Architect helps you to model Spatial Reference information for your ArcGIS schema, including the selection of predefined coordinate systems and associated values.	Setting ArcGIS Coordinate Systems [1637]
Disable MDG Technology for ArcGIS	If you prefer not to use the MDG Technology for ArcGIS in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).	MDG Technologies [1285]

8.5.1.1 ArcGIS Toolbox Pages

The ArcGIS Toolbox pages provide elements and connectors that enable you to model ArcGIS geodatabase concepts and relationships.

Access [View | Diagram Toolbox: More Tools | ArcGIS](#)

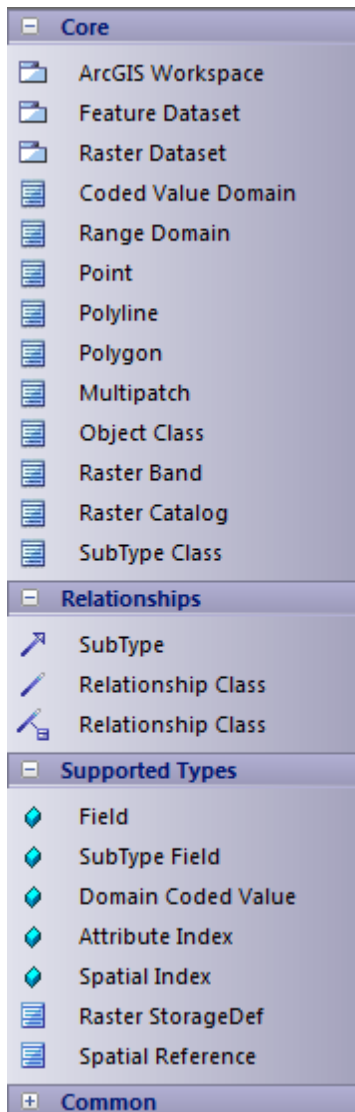


Diagram toolbox icons:

Toolbox Icon	Description
Elements	
ArcGIS Workspace	The geodatabase workspace package, which holds all the ArcGIS modeling elements. Export the contents of this package to produce the Geodatabase XML Workspace Document, which can be imported to <i>Esri ArcCatalog</i> .
Feature Dataset	A stereotyped package that holds or organizes the Point, Polyline, Polygon or Multipatch elements with the same spatial reference. The Feature Dataset is only created under the ArcGIS Workspace package; it can not be created under another Feature Dataset package.

Toolbox Icon	Description
Raster Dataset	A stereotyped package that holds or organizes the raster data (as Raster Band elements; see below).
Coded Value Domain	An extended UML Class, representing a set of valid values that might apply to any type of attribute.
Range Domain	An extended UML Class, representing a valid range of numeric values that might apply to a numeric type of attribute.
Point	An extended UML Class, representing the ArcGIS Point.
Polyline	An extended UML Class, representing the ArcGIS Polyline.
Polygon	An extended UML Class, representing the ArcGIS Polygon.
Multipatch	An extended UML Class, representing the ArcGIS Multipatch.
Object Class	An extended UML Class, representing a collection of nonspatial data of the same type or Class.
Raster Band	An extended UML Class, representing one layer of a matrix of cell values. Every Raster Dataset contains one or more Raster Bands.
Raster Catalog	An extended UML Class, representing a collection of Raster Datasets in the geodatabase.
SubType Class	An extended UML Class, holding a subset of the attributes of an element in the Feature Dataset.
Relationships	
Subtype	An extended UML Association, providing the relationship between a Feature Class element and a SubType Class element.
RelationshipClasses	An extended UML <i>Association</i> , providing the relationship between: <ul style="list-style-type: none"> • Two elements in the Feature Dataset, or • An element in the Feature Dataset and an Object Class element

Toolbox Icon	Description
RelationshipClasses	An extended UML <i>Association Class</i> , providing the attributed relationship between: <ul style="list-style-type: none"> Two elements in the Feature Dataset, or An element in the Feature Dataset and an Object Class element
Supported Types	
Field	An extended UML attribute that represents an ArcGIS field of the geodatabase, in a table or feature class.
SubType Field	An extended UML Attribute that represents the subtype field of an ArcGIS table or feature class.
DomainCodedValue	An extended UML Attribute that specifies the value of an ArcGIS Coded Value Domain.
Attribute Index	An extended UML Attribute that represents the ArcGIS Attribute Index.
Spatial Index	An extended UML Attribute that represents the ArcGIS Spatial Index.
Raster StorageDef	An extended UML Class, representing the storage properties for a Raster value in the geodatabase; this information is required when a Raster Dataset element is created.
Spatial Reference	An extended UML Class that defines the spatial reference information of your schema, such as a coordinate system and XYTolerance. You can define one or more Spatial Reference elements, which you link to other ArcGIS elements via their <i>Spatial Reference</i> Tagged Value.

8.5.1.2 Setting ArcGIS Coordinate Systems

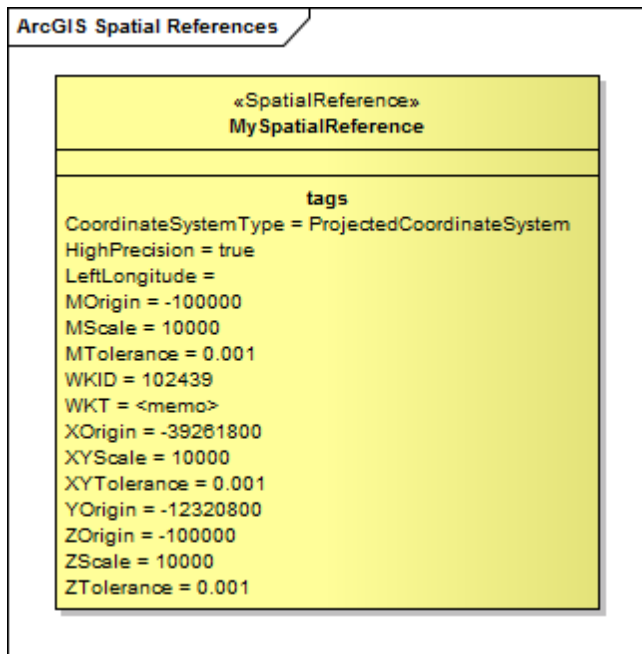
ArcGIS Feature Classes and Feature Datasets use *spatial references*, which consist of a coordinate system and associated values such as XY resolution and various tolerance values.

Enterprise Architect enables you to capture spatial reference properties using a Class stereotyped as *SpatialReference*, which is available from the ArcGIS Toolbox pages. The ArcGIS model pattern includes a package named *Spatial References*, as a placeholder for creating such elements.

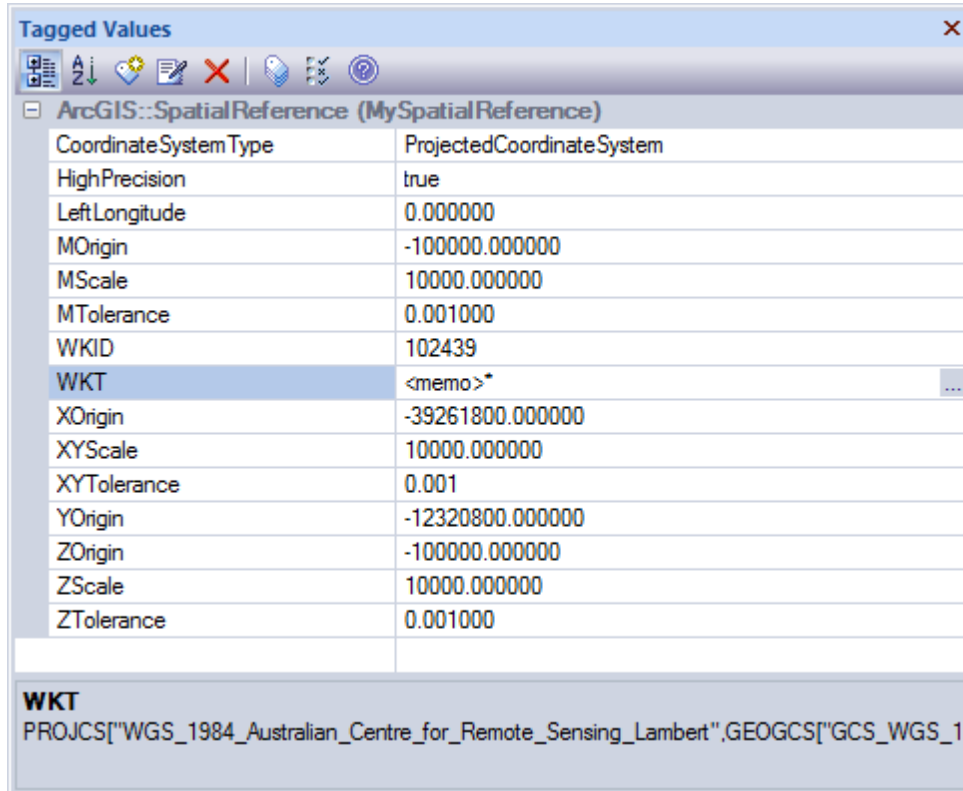
To help you model spatial reference properties, Enterprise Architect provides a user interface for selecting one of the predefined coordinate systems. When you select a Geographic or Projected coordinate system, Enterprise Architect automatically inserts default values for the associated properties, such as Well Known Text (WKT), resolution, precision or tolerances. These values are held as Tagged Values on the *SpatialReference* element.

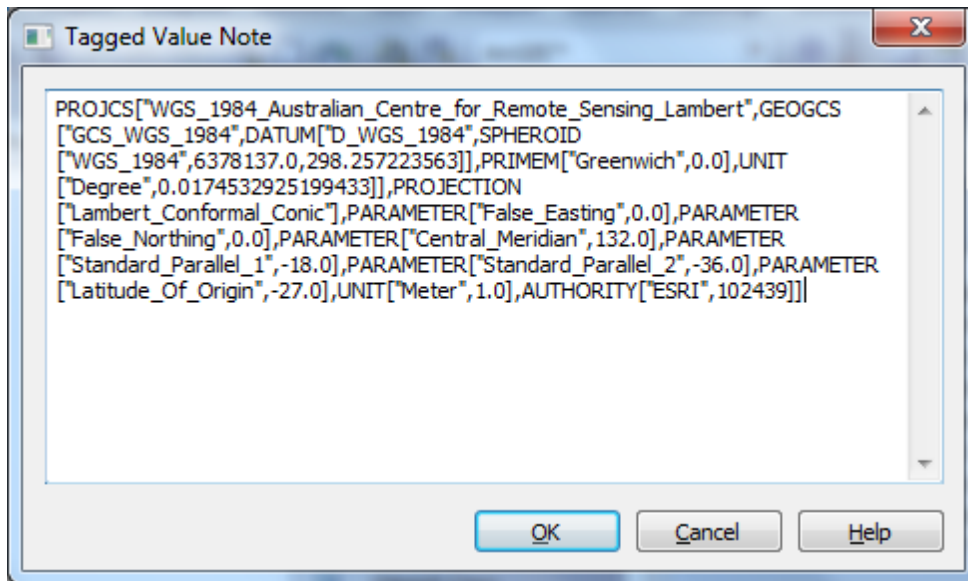
You can refer to a Spatial Reference Class from any Feature Dataset or Feature Class in your ArcGIS model. Each Feature Dataset and Feature Class has a Tagged Value named *SpatialReference* for this purpose.

Below is an example *SpatialReference* element.



As shown by the Tagged Value window below, a Projected Coordinate system has been selected, specifically *WGS 1984 Australian Centre for Remote Sensing Lambert*.





How to

To define a Spatial Reference element in Enterprise Architect

Step	Action	See also
1	Open the diagram under the <i>Spatial References</i> package of your ArcGIS model. (You can actually use any ArcGIS diagram in your model to define Spatial Reference elements; however, the above diagram is a convenient placeholder created by Enterprise Architect's model pattern for ArcGIS.)	
2	Drag a Spatial Reference element from the ArcGIS Toolbox onto the diagram.	
3	Click on the Spatial Reference element.	
4	Select the Extensions ArcGIS Set Coordinate System menu option.	
5	Select one of the listed coordinate systems.	
6	Click on the OK button.	

Notes

- You can refer to this element from any other Feature Dataset or Feature Class, using the *SpatialReference* Tagged Value; the Spatial Reference element thus provides a single-point of control, should you need to change the Spatial Reference information later

8.5.2 Export ArcGIS XML Workspace

You can export your modeled Geodatabase Workspace XML Document (containing the ArcGIS schema), which can be imported to *Esri ArcCatalog*. To export a Geodatabase Workspace XML Document, use the *Publish Model Package* facility.

Access Click on an ArcGIS stereotyped package (your ArcGIS Workspace Package) in the Project Browser, then select either:

Extensions | Publish | ArcGIS

Extensions | ArcGIS | Export to ArcGIS Workspace XML

Project | Model Publisher

Project | Model Import/Export | Export Package to XMI File: Publish, or

Right-click | Import/Export | Export Package to XMI: Publish

Use to

- Export your modeled Geodatabase Workspace XML Document

Reference

On the Publish Model Package dialog:

Field/Option/ Button	Usage	See also
Root Package	Displays the name of the selected ArcGIS Workspace package.	
Filename	Type in or browse for the file path into which the XML file is to be generated.	
XML Type	Select ArcGIS as the XML/XMI version to export the package to.	
Format XML Output	Format the output into readable XML (this takes a few more seconds at the end of the run).	
Write Log File	Write a log of the export activity (recommended). The log file is saved to the directory into which the XML file is exported.	
View XML	View the exported XML file.	
Export	Initiate XML export.	
Close	Close this dialog.	

Progress	Indicates the progress of the XML export.	
-----------------	---	--

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Export XMI** permission to export to XML
- Before exporting your model to an ArcGIS schema, you must define at least one Spatial Reference element; Spatial Reference elements are referred to by other schema elements via a dynamically linked Tagged Value, named *SpatialReference*

Learn more

- [Geodatabase Design for ArcGIS](#)^[1635]
- [Publish Model Package](#)^[414]
- [List of Available Permissions](#)^[264]
- [Import ArcGIS XML Workspace](#)^[1647]
- [Setting ArcGIS Coordinate Systems](#)^[1637]

8.5.3 Import ArcGIS XML Workspace

Enterprise Architect can import a Geodatabase Workspace XML Document (containing the ArcGIS schema) from the Esri *ArcCatalog*.

Access Click on the target package in the Project Browser, then select one of the following:

Extensions | Import | ArcGIS

Extensions | ArcGIS | Import ArcGIS Workspace XML

Project | Model Import/Export | Import Package From XMI File: Other XML File Formats | ArcGIS

Right-click | Import/Export | Import Package From XMI File: Other XML File Formats | ArcGIS

Use to

- Import a Geodatabase Workspace XML Document into an Enterprise Architect ArcGIS model

Reference

On the Import Package from ArcGIS XML dialog:

Field/Option/Button	Action	See also
Filename	Type in or browse for the name of the ArcGIS XML file to import.	
Create Diagrams	Select the checkbox to create Class diagrams under the imported packages.	

Field/Option/Button	Action	See also
Strip GUIDs	The Strip GUIDs feature is currently mandatory for ArcGIS imports, which means that elements are created 'as new' each time an ArcGIS schema is imported.	
Write Log File	Select to write a log of import activity (recommended). The log file is saved in the directory from which the file is being imported, with the same name as the imported file plus the suffix <i>_import.log</i> .	
View XML	Click to view the XML before import.	
Import	Click to import the ArcGIS XML file.	
Close	Click to close this dialog.	
Help	Click to display this Help page.	
Import Progress	Indicates the progress of the import.	

Learn more

- [Geodatabase Design For ArcGIS](#) ^[1633]
- [ArcGIS UML Profile](#) ^[1633]
- [Export ArcGIS XML Workspace](#) ^[1640]

8.6 MDG Technology for ODM

MDG Technology for ODM is the implementation of the Object Management Group's **Ontology Definition Metamodel** for Enterprise Architect. It provides ontology modeling capabilities within Enterprise Architect (9.2 or later), enabling you to develop large-scale ontologies within the fully-integrated modeling environment, for your project domain.

Through the MDG Technology for ODM, you have access to the following features:

- A **UML Profile for the Resource Description Framework (RDF)**
- A **UML Profile for Web Ontology Language (OWL)**
- Customized **diagram types** and **toolbox pages**, for convenient access to elements and relationships to model ontologies and resources effectively
- **Model Templates** to get started quickly with ontology modeling
- Commands to **import and export** RDF and OWL files, and to define new **namespaces** for ODM packages and new **labels** for OWL and RDF elements

Topics

Topic	Detail	See also
Ontologies	<p>Ontologies define a common vocabulary for sharing information in a domain. They provide a standardized, machine-interpretable definition of basic concepts within the domain, and of relations among them. Formally defined ontologies give stakeholders the ability to:</p> <ul style="list-style-type: none"> • Share common understanding of the structure of information among themselves or among software agents • Reuse domain knowledge • Explicitly codify domain assumptions • Separate domain knowledge from operational knowledge • Analyze domain knowledge effectively 	
The OMG ODM	<p>The Ontology Definition Metamodel has been formalized by the OMG as a standard.</p> <p>It defines an important set of enabling capabilities for Model Driven Architecture (MDA) for a formalized representation of business semantics and taxonomies, and knowledge representation based on those formalizations. It defines extensions to the Unified Modeling Language (UML) to provide a custom modeling notation for ontology definition. These extensions are:</p> <ul style="list-style-type: none"> • The Resource Description Framework (RDF), used to represent data, properties and formal semantics of information in the web 	

Topic	Detail	See also
	<ul style="list-style-type: none"> Web Ontology Language (OWL), used to represent terms in vocabularies, and their interrelationships; OWL extends the RDF to define both the domain information and the relevant domain meaning 	
Disable MDG Technology for ODM	If you prefer not to use the MDG Technology for ODM in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).	

Learn more

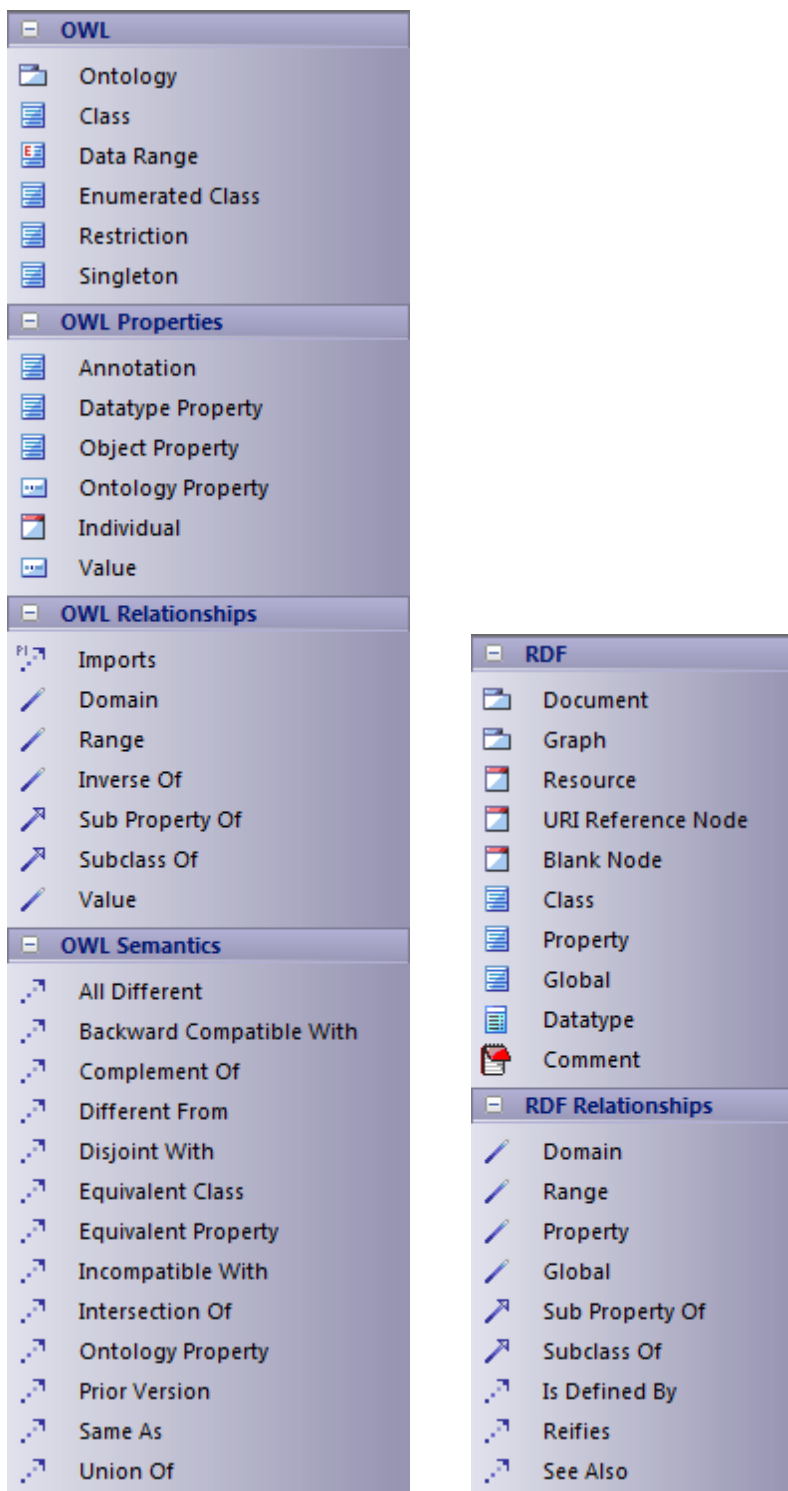
- [Example ODM Diagrams](#) ^[1651]
- [ODM Toolbox Pages](#) ^[1644]
- [ODM Commands](#) ^[1652]
- [New Diagram](#) ^[724]
- [Quick Linker](#) ^[791]
- [Model Wizard](#) ^[660]
- [Toolbox Shortcut](#) ^[704]

8.6.1 ODM Toolbox Pages

Enterprise Architect provides two sets of toolbox pages for ODM:

- Web Ontology Language (OWL) pages for Ontology Definition diagrams and Ontology Facts diagrams, and
- Resource Description Framework (RDF) pages for Resource Definition diagrams

Access [View | Diagram Toolbox: More tools | ODM | Web Ontology Language, and View | Diagram Toolbox: More tools | ODM | Resource Description Framework](#)



Learn more

- [OWL Elements & Relationships](#) ¹⁶⁴⁶
- [RDF Elements & Relationships](#) ¹⁶⁴⁹
- [Example ODM Diagrams](#) ¹⁶⁵¹

8.6.1.1 OWL Elements & Relationships

This topic explains each of the icons on the Web Ontology Language (OWL) Toolbox pages for Ontology Definition diagrams and Ontology Facts diagrams.

Access [View | Diagram Toolbox: More tools | ODM 1.0 | Web Ontology Language](#)

OWL Elements:

Toolbox Icon	Description
Ontology	The OWL ontology package, which holds all the OWL modeling elements. You can export the contents of this package to produce the Ontology XML document.
Class	An extended UML Class that represents an OWL Class that describes an instantiable entity with properties and semantic meaning.
Data Range	An extended UML Enumeration that defines a collection of values for an OWL Property.
Enumerated Class	An extended UML Class that defines an OWL Class extension defined by any one of the range of the allowed OWL Individuals.
Restriction	An extended UML Class that defines an OWL Class extension as restricted by the specified property and its allowable values.
Singleton	An extended UML Class, representing an OWL Class for a specific OWL Individual.

OWL Properties (Elements):

Toolbox Icon	Description
Annotation	An extended UML Class, representing an OWL Annotation Property definition.
Datatype Property	An extended UML Class, representing an OWL Datatype Property definition.
Object Property	An extended UML Class, representing a semantic OWL Property definition.
Ontology Property	An extended UML Part, representing a property defined on the OWL Ontology.

Toolbox Icon	Description
Individual	An extended UML Object, representing an instance of an OWL Class which defines an individual fact.
Value	An extended UML Part, holding a value defined in an OWL Property or OWL Individual.

OWL Relationships:

Toolbox Icon	Description
Imports	An extended UML PackageImports, that enables an OWL ontology to reference another OWL Ontology.
Domain	An extended UML Association, that specifies the OWL Classes that apply the specified OWL Property (Annotation, Datatype or Object Property).
Range	An extended UML Association, that specifies the OWL Class with the value type applicable to the specified OWL Property (Annotation, Datatype or Object Property).
Inverse Of	An extended UML Association, between two opposing, but related OWL Property elements.
Sub Property Of	An extended UML Generalization between two OWL Property elements.
Subclass Of	An extended UML Generalization between two OWL Class elements.
Value	An extended UML Association, defining an OWL Property and value between OWL Classes.

OWL Semantics (Relationships):

Toolbox Icon	Description
All Different	An extended UML Dependency between different (ie: unique) OWL Individuals typed by the same OWL Class.

Toolbox Icon	Description
Backward Compatible With	An extended UML Dependency between an OWL Ontology and another that it's backward compatible with.
Complement Of	An extended UML Dependency between an OWL Class and its complement (or opposite).
Different From	An extended UML Dependency between two semantically different OWL Individuals typed by the same OWL Class.
Disjoint With	An extended UML Dependency between two OWL Classes that have no common individuals.
Equivalent Class	An extended UML Dependency between two equivalent OWL Classes.
Equivalent Property	An extended UML Dependency between two equivalent OWL Property elements.
Incompatible With	An extended UML Dependency between an OWL Ontology and another that it is incompatible with.
Intersection Of	An extended UML Dependency between an OWL Class and others it also specializes.
Ontology Property	An extended UML Dependency, representing a property defined on the OWL Ontology.
Prior Version	An extended UML Dependency between an OWL Ontology and its predecessor.
Same As	An extended UML Dependency between two semantically identical OWL Individuals typed by the same OWL Class.
Union Of	An extended UML Dependency between a general OWL Class and others that distinctly specialize it.

[Learn more](#)

- [ODM Toolbox Pages](#)  ¹⁶⁴⁴

8.6.1.2 RDF Elements & Relationships

This topic explains each of the icons on the Resource Description Framework (RDF) pages for Resource Definition diagrams.

[Access](#) [View](#) | [Diagram Toolbox: More tools](#) | [ODM 1.0](#) | [Resource Description Framework](#)

RDF Elements:

Toolbox Icon	Description
Document	The RDF Document package, which holds all the RDF modeling elements. You can export the contents of this package to produce the Resource Description XML document.
Graph	An extended UML Package that represents a set of RDF subject and object triples within the RDF Document.
Resource	An extended UML Object that represents a uniquely identifiable general resource.
URI Reference Node	An extended UML Object that represents a uniquely identifiable external resource.
Blank Node	An extended UML Object that represents a uniquely identifiable internal resource.
Class	An extended UML Class representing an RDF Class, which describes an instantiable resource with properties.
Property	An extended UML Class, representing an RDF Property definition.
Global	An extended UML Class, representing a global RDF Property definition.
Datatype	An extended UML Datatype, representing an RDF Datatype definition.
Comment	A UML Comment element.

RDF Relationships:

Toolbox Icon	Description
Domain	An extended UML Association that specifies the RDF Classes that apply the

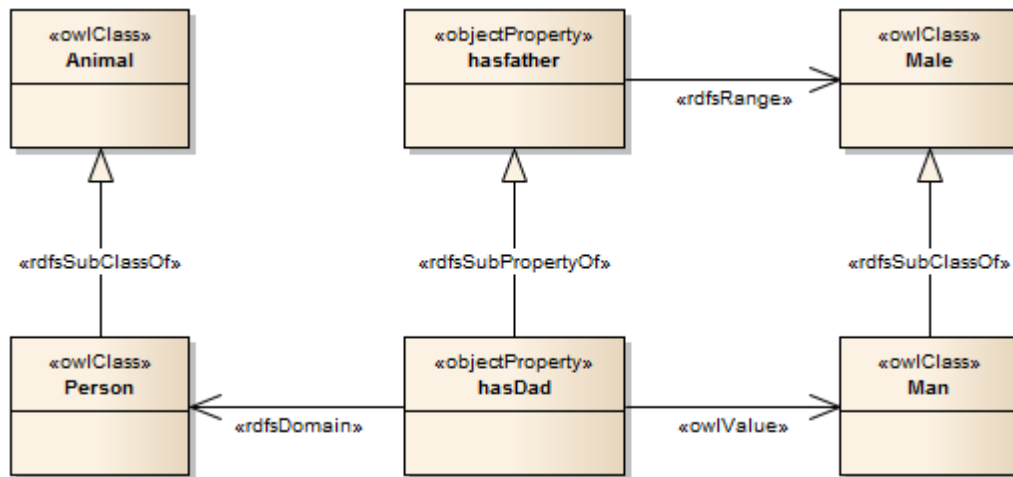
Toolbox Icon	Description
	specified RDF Property.
Range	An extended UML Association that specifies the RDF Class with the value type applicable to the specified RDF Property.
Property	An extended UML Association that defines a RDF property between two RDF Classes.
Global	An extended UML Association that defines a global RDF property between two RDF Classes.
Sub Property Of	An extended UML Generalization between two RDF Property elements.
Subclass Of	An extended UML Generalization between two RDF Class elements.
Is Defined By	An extended UML Dependency between a RDF Resource and another that defines it.
Reifies	An extended UML Dependency between a RDF Resource and another that it reifies.
See Also	An extended UML Dependency between a RDF Resource and another that contains more information about it.

Learn more

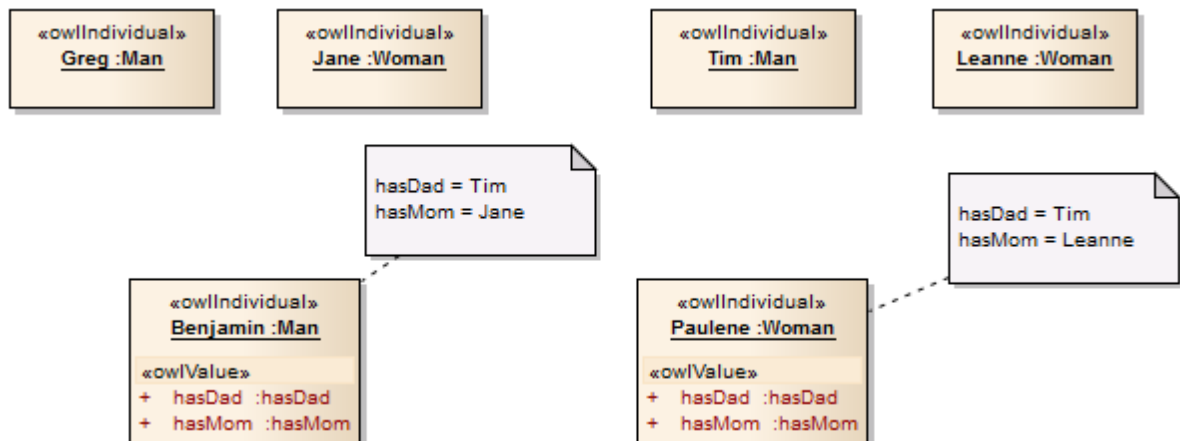
- [ODM Toolbox Pages](#) 

8.6.2 Example ODM Diagrams

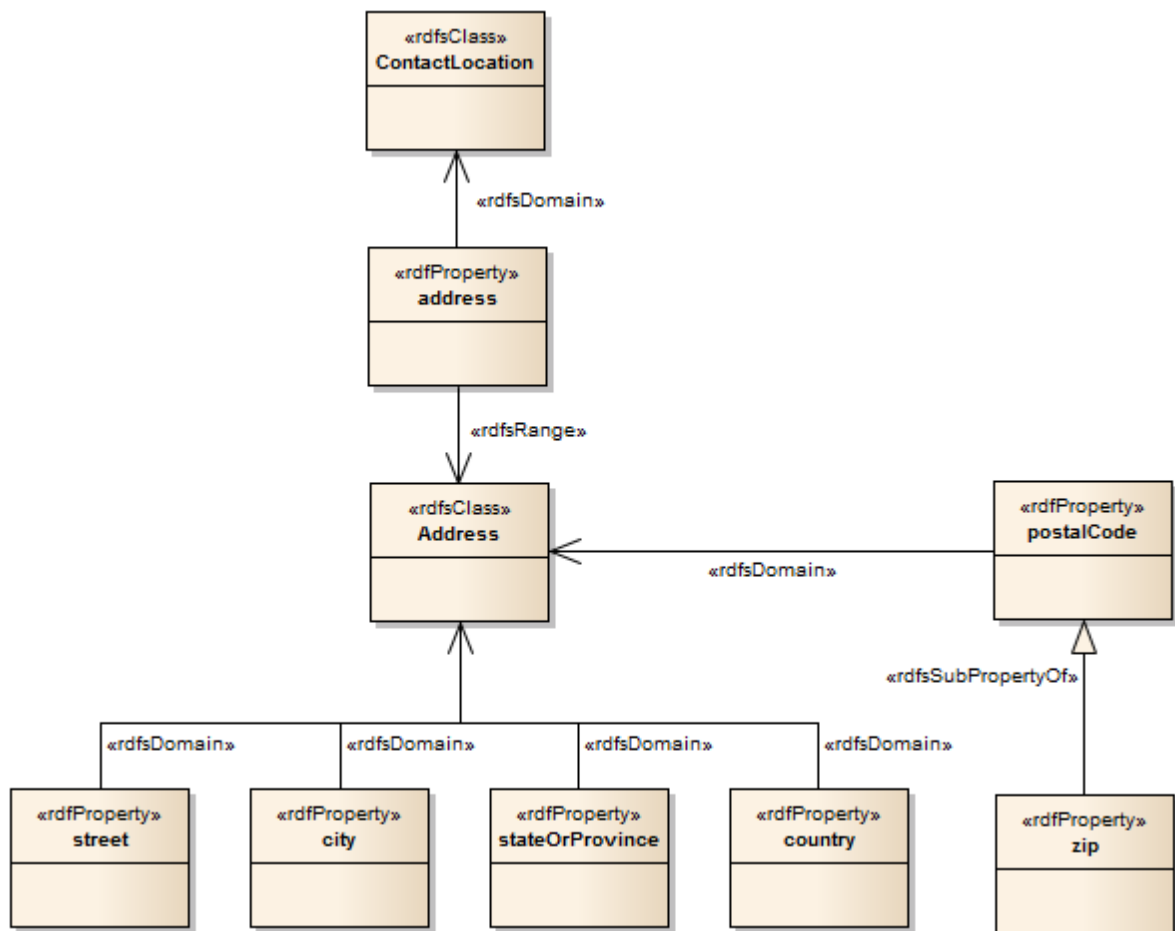
Example OWL Definition Diagram:



Example OWL Facts Diagram:



Example RDF Definition Diagram:



Learn more

- [MDG Technology for ODM](#)^[1643]
- [ODM Toolbox Pages](#)^[1644]

8.6.3 ODM Commands


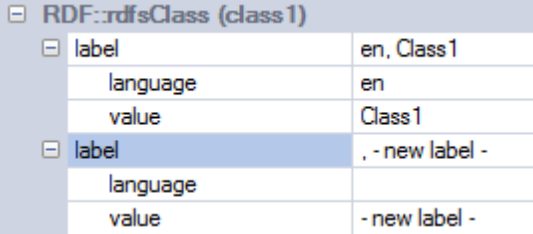

The MDG Technology for ODM provides four management commands to help you maintain your ODM models:

- **Export** OWL/RDF models as an .rdf or .xml file
- **Import** OWL or RDF content (in .owl or .rdf xml file format) as an ODM model package
- Create a new **namespace definition** for an rdfDocument or owlOntology package
- Create a new **label definition** for a valid RDF or OWL element (such as an owlClass, objectProperty, rdfClass or rdfProperty)

Access [Extensions | ODM | <command>](#)
[Extensions | Import | OWL/RDF, or](#)
[Extensions | Publish | OWL/RDF](#)

Commands:

Command	Detail	See also																		
Export OWL/RDF	<p>In the Project Browser, select the rdfDocument or owlOntology package to export.</p> <p>Select either:</p> <ul style="list-style-type: none"> • Extensions ODM Export OWL/RDF, or • Extensions Publish OWL/RDF <p>A browser dialog displays, on which you specify the file location to export to, and the .owl or .rdf filename.</p> <p>Click on the Save button.</p> <p>The package content is exported to the specified file.</p>																			
Import OWL/RDF	<p>In the Project Browser, select the rdfDocument or owlOntology package into which to import the .owl or .rdf file.</p> <p>Select either:</p> <ul style="list-style-type: none"> • Extensions ODM Import OWL/RDF, or • Extensions Import OWL/RDF <p>A browser dialog displays, through which you locate and select the file to import.</p> <p>Click on the Open button.</p> <p>The file content is imported into the selected package.</p>																			
Define New Namespace	<p>Ensure that the Show Duplicate Tags checkbox is selected in the Options dialog (Tools Options Objects).</p> <p>In the Project Browser, select the RDF or OWL package to be allocated a new namespace.</p> <p>Select the Extensions ODM Define New Namespace command.</p> <p>Double-click on the selected package to display the package Properties dialog, and select the RDF page. Notice the additional <i>namespaceDefinition</i> tag.</p> <table border="1"> <tbody> <tr> <td><input type="checkbox"/></td> <td>namespaceDefinition</td> <td>ns0, http://myontologies/newOntology</td> </tr> <tr> <td></td> <td>namespacePrefix</td> <td>ns0</td> </tr> <tr> <td></td> <td>namespaceURI</td> <td>http://myontologies/newOntology</td> </tr> <tr> <td><input type="checkbox"/></td> <td>namespaceDefinition</td> <td>- new prefix -, - new URI -</td> </tr> <tr> <td></td> <td>namespacePrefix</td> <td>- new prefix -</td> </tr> <tr> <td></td> <td>namespaceURI</td> <td>- new URI -</td> </tr> </tbody> </table> <p>Click on the namespacePrefix and namespaceURI value fields for the</p>	<input type="checkbox"/>	namespaceDefinition	ns0, http://myontologies/newOntology		namespacePrefix	ns0		namespaceURI	http://myontologies/newOntology	<input type="checkbox"/>	namespaceDefinition	- new prefix -, - new URI -		namespacePrefix	- new prefix -		namespaceURI	- new URI -	
<input type="checkbox"/>	namespaceDefinition	ns0, http://myontologies/newOntology																		
	namespacePrefix	ns0																		
	namespaceURI	http://myontologies/newOntology																		
<input type="checkbox"/>	namespaceDefinition	- new prefix -, - new URI -																		
	namespacePrefix	- new prefix -																		
	namespaceURI	- new URI -																		

Command	Detail	See also
	<p>new namespace definition, and type in the appropriate new values.</p> <p>If necessary, click on the previous <i>namespaceDefinition</i> tag and on the  (Delete) icon in the RDF page toolbar.</p> <p>Click on the OK button.</p>	
Define New Label	<p>Ensure that the Show Duplicate Tags checkbox is selected in the Options dialog (Tools Options Objects).</p> <p>In the Project Browser, select the RDF or OWL element to be allocated a new label.</p> <p>Select the Extensions ODM Define New Label command.</p> <p>Double-click on the selected element to display the element Properties dialog, and select the RDF page. Notice the additional <i>label</i> tag.</p>  <p>Click on the language and value value fields for the new label definition, and type in the appropriate new values.</p> <p>If necessary, click on the previous <i>label</i> tag and on the  (Delete) icon in the RDF page toolbar.</p> <p>Click on the OK button.</p>	

Learn more

- [MDG Technology for ODM](#)^[1643]









8.7 User Interface Models

User Interface Diagrams are customized UML diagrams used to visually mock-up a system's user interface using forms, controls and labels.

Example Diagram [Example User Interface Diagram](#)^[1656]

Tools

Select User Interface diagram elements and connectors from the User Interface pages of the Toolbox.

User Interface Diagram Elements	User Interface Diagram Connectors
 Package	 Associate
 Screen	 Aggregate
 UI Control	 Generalize
 Object	 Realize

Notes

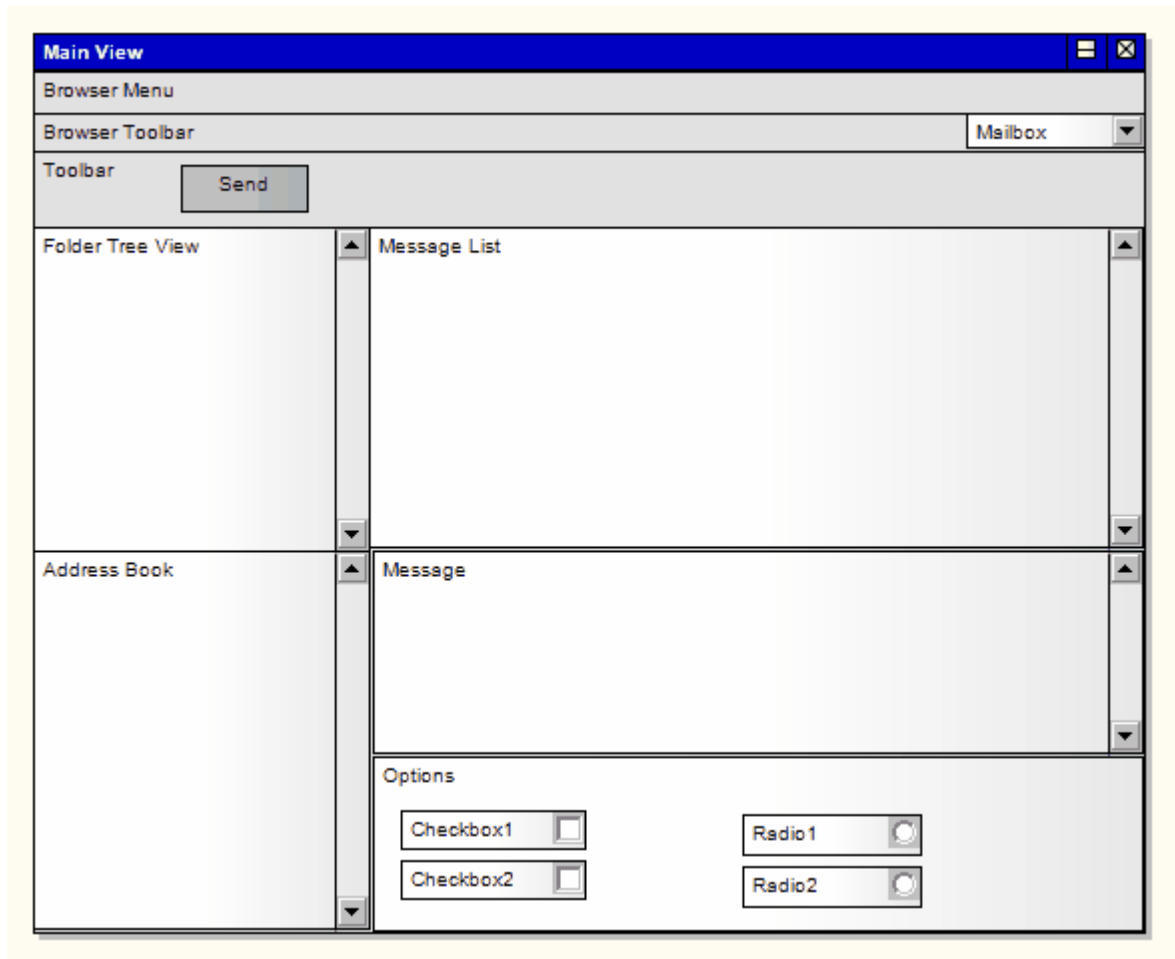
- Using stereotyped Classes, you can model the design of a web page user interface
- The Enterprise Architect Professional, Corporate and Suite editions also include the MDG Win32 UI Technology, which enables you to design user interface components that render more precisely as Win32® User Interface elements

Learn more

- [Web Stereotypes](#)^[1660]
- [Win32 UI Technology](#)^[1661]

8.7.1 Example User Interface Diagram

In the example User Interface diagram below, forms, controls and labels are arranged on the diagram to describe its appearance. UI Control elements can also be traced to other model elements linking the UI with the underlying implementation.



Learn more

- [Screen](#)^[1656]
- [UI Control Elements](#)^[1657]

8.7.2 Screen

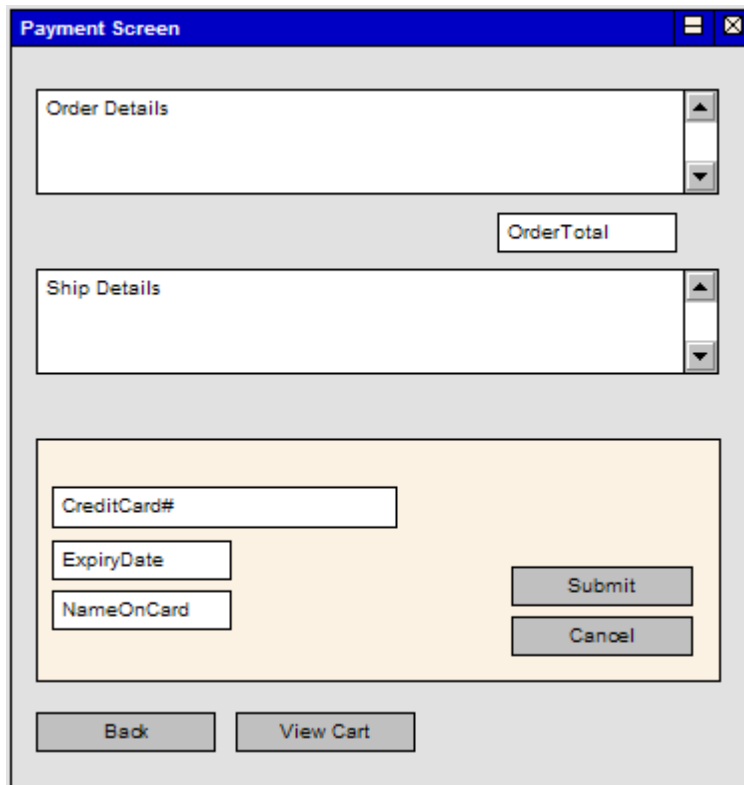
Description

A Screen is used to prototype User Interface screen flow. By using UML features such as requirements, constraints and scenarios against User Interface diagram elements, you can build up a solid and detailed understanding of user interface behavior without having to use code. This becomes an excellent means of establishing the precise behavior the system has from a user perspective, and in conjunction with the Use Case model, defines exactly how a user gets work done.

Web pages can also be prototyped and specified rigorously using Enterprise Architect's custom interface extensions.

Example

The example diagram below illustrates some features of Enterprise Architect's screen modeling extensions that support web page prototyping. By adding requirements, rules, scenarios and notes to each element, a detailed model is built up of the form or web page, without having to resort to GUI builders or HTML.



Enterprise Architect displays [UI Controls](#)¹⁶⁵⁷ as a range of special icons, depending on the stereotype used; for example, a Control stereotyped as a «list» displays with a vertical scroll bar.

Toolbox icon



Learn more

- [User Interface Models](#)¹⁶⁵⁵
- [UI Controls](#)¹⁶⁵⁷

8.7.3 UI Control Elements

Description

A UI Control element represents a user interface control element (such as an edit box). It is used for

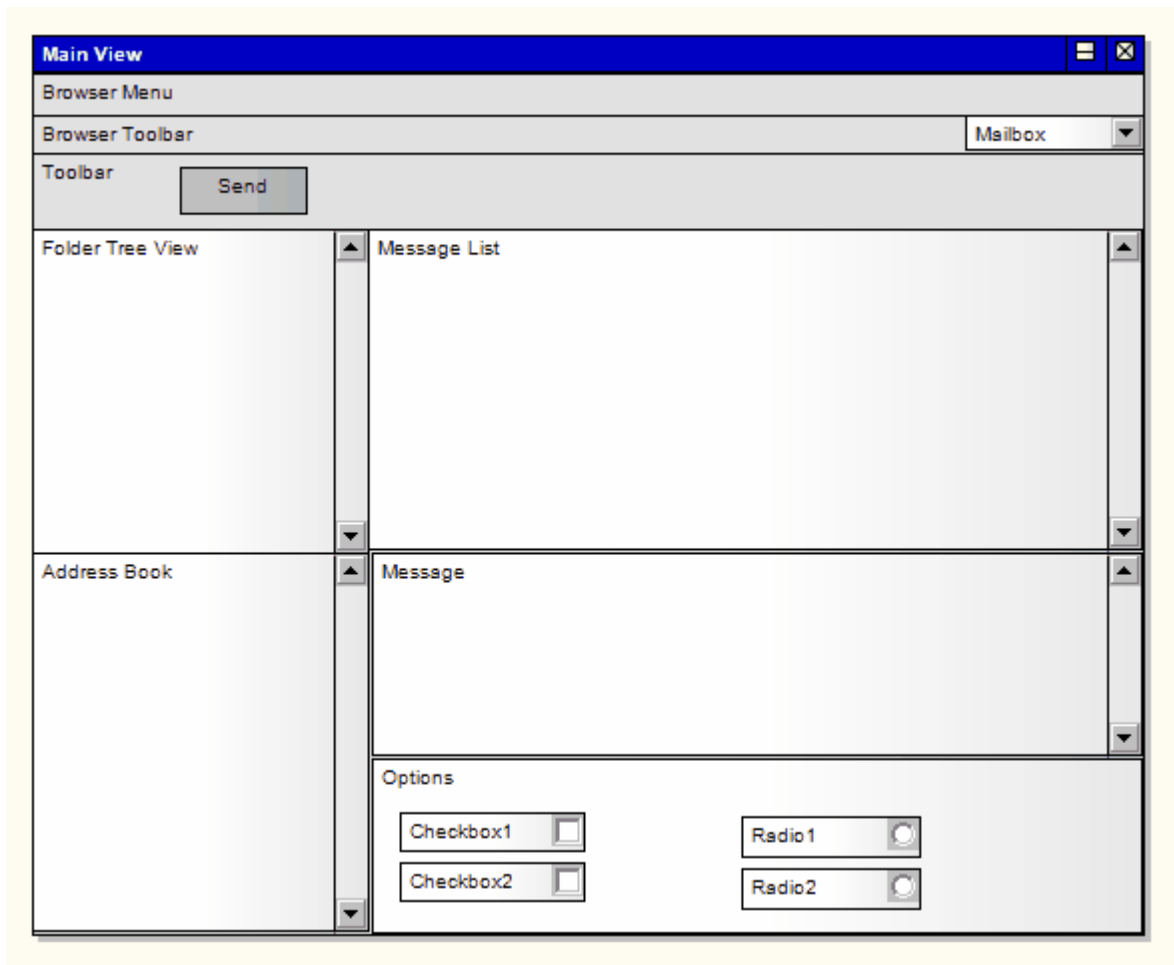
capturing the components of a screen layout and requirements in a Custom or User Interface diagram.

There are a number of UI Control elements available in the User Interface page of the Toolbox. These include:

- List
- Table
- Text Box
- Label
- Form
- Panel
- Button
- Combobox
- Checkbox
- Checkbox (left hand side)
- Radio button
- Radio button (left hand side)
- Vertical Line
- Horizontal Line

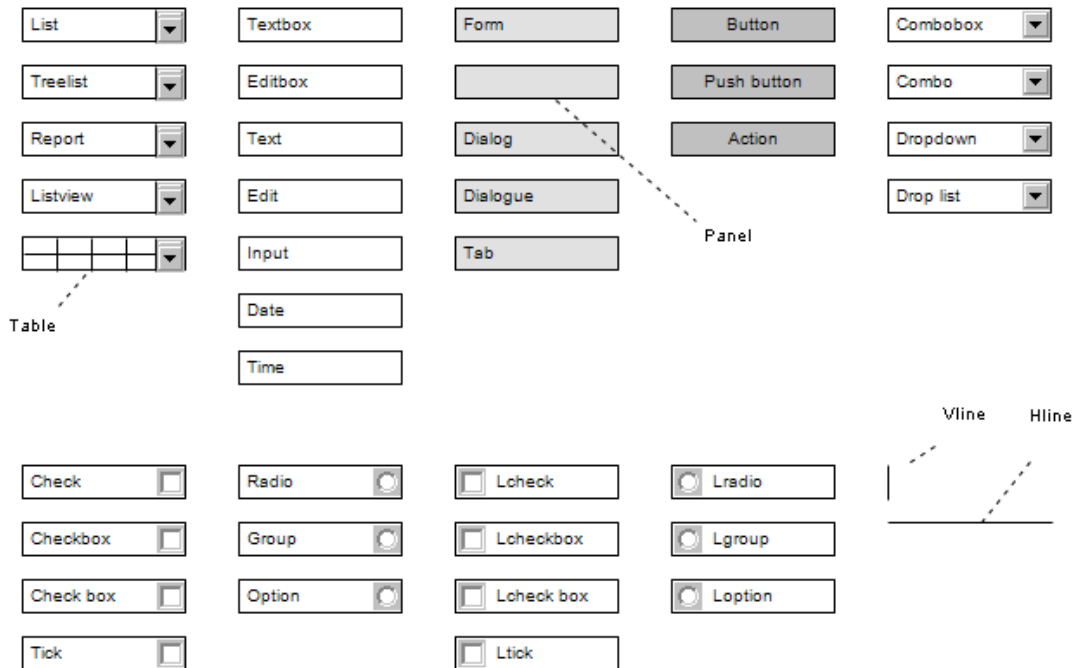
Example

The icons can be combined on a Screen icon to represent the appearance of a user interface screen, as shown:

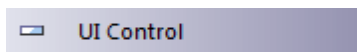


You can also extend the available icons by selecting other stereotypes in the UI Control Element Properties dialog. The full set of available stereotypes is shown below; type or select the text in the **Stereotype** field to create the corresponding icon.

ui User Interface



Toolbox icon



(where **UI Control** is the name of the user interface element type)

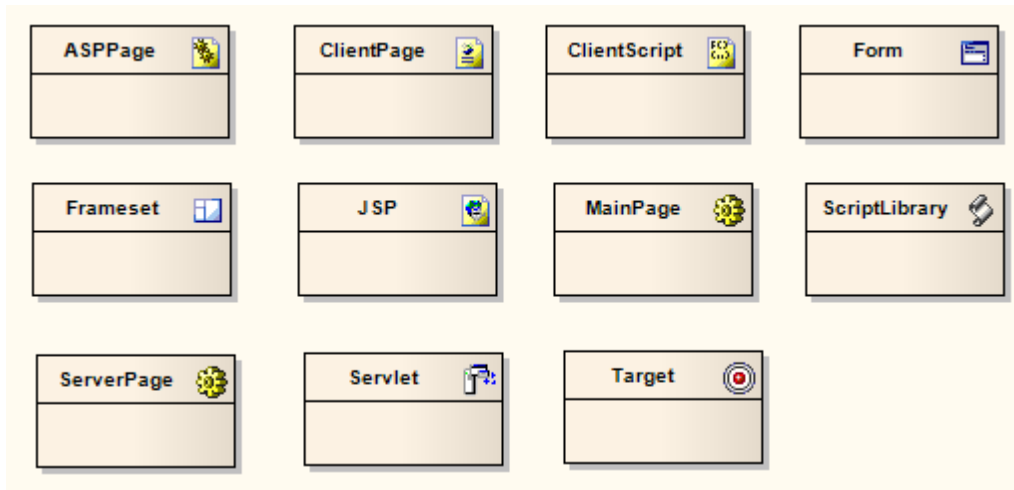
Learn more

- [User Interface Models](#) ^[1655]
- [Screen](#) ^[1656]

8.7.4 Web Stereotypes

Description

Enterprise Architect supports a number of stereotypes for web page modeling, the graphical elements for which display with a **graphical icon** instead of the usual «stereotype» format. These stereotypes are only supported for Class elements. The image below indicates the various graphical icons and their associated stereotypes.



A similar set of web modeling elements and their relationships are also available through dedicated Web Modeling pages in the Diagram Toolbox.

How to

To set a web icon

Step	Action
1	Create a new Class element in a diagram.
2	Display the Class Properties dialog.
3	In the Stereotype field, either type in the required stereotype name or click on the drop-down arrow and select the required stereotype (as named above).
4	Click on the OK button. The Class displays as in one of the examples above.

Learn more

- [User Interface Models](#)^[1655]
- [Diagram Toolbox](#)^[698]

8.7.5 Win32 UI Technology

The MDG Win32 UI Technology enables you to design user interface screens that render as Win32® user interface elements.

Use to

- Create and maintain User Interface diagrams in the Win32® User Interface formats

Access The Win32 UI facilities are provided in the form of:

- A Win32 User Interface diagram type, accessed through the New Diagram dialog
- Win32® User Interface pages in the Toolbox
- Win32 User Interface element and relationship entries in the Toolbox Shortcut menu and Quick Linker

Win32 User Interface Toolbox Pages:

You can access the Win32 User Interface Toolbox pages through the **View | Diagram Toolbox: More tools... User Interface - Win32** option.

You can also set Win32® User Interface Modeling as the active default technology to access the Toolbox pages directly.

Disable Win32 User Interface Technology:

If you prefer not to use the Win32® UI Technology in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (**Settings | MDG Technologies**).

Notes

- The MDG Win32® User Interface Technology is available in the Enterprise Architect Professional, Corporate and Suite editions

Learn more

- [User Interface Models](#) 

Learning Center topics

- **(Alt + F1) | Modeling Languages | Standard Profiles | Win32 User Interface**
- **(Alt + F1) | Software Engineering | User Interface | Win32 Dialog Modeling**

8.8 Other Stereotypes

There are many other UML elements that you can also work with in Enterprise Architect, most of which are basic elements extended by the use of stereotypes. This topic gives a brief introduction to some of these elements.

- [Analysis Stereotypes](#)^[1508]
- [Boundary](#)^[1663]
- [Composite Elements](#)^[825]
- [Control](#)^[1665]
- [Entity](#)^[1666]
- [Event](#)^[1667]
- [Feature](#)^[1485]
- [Hyperlinks](#)^[1668]
- [Image](#)^[1672]
- [N-Ary Association](#)^[1673]
- [Packaging Component](#)^[1674]
- [Process](#)^[1674]
- [Requirement](#)^[1471]
- [Risk](#)^[1675]
- [Screen](#)^[1656]
- [Task](#)^[1676]
- [Test Case](#)^[1677]
- [Table](#)^[1630]
- [UI Control Elements](#)^[1657]
- [Web Stereotypes](#)^[1660]

Learn more

- [UML Stereotypes](#)^[1264]

8.8.1 Boundary

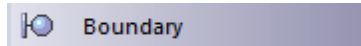


Description

A Boundary is a stereotyped Object that models some system boundary, typically a user interface screen. You can also create a Boundary as a stereotyped Class. See the *Create a Boundary* topic. Boundary elements are used in analysis to capture user interactions, screen flows and element interactions (or 'collaborations').

A Boundary is used in the conceptual phase to capture users interacting with the system at a screen level (or some other boundary interface type). It is often used in Sequence and Robustness (Analysis) diagrams. It is the View in the Model-View-Controller pattern.

Toolbox icon



Learn more

- [Create a Boundary](#)^[1664]
- [Analysis Stereotypes](#)^[1508]
- [Sequence Diagram](#)^[1074]
- [Communication Diagram](#)^[1084]
- [Object Diagram](#)^[1015]
- [Analysis Diagram](#)^[1509]

8.8.1.1 Create a Boundary

To create a Boundary element on a diagram as an Object, follow the steps below

Step	Action
1	In the Toolbox, select the More tools Analysis menu option.
2	From the Analysis Elements page, drag the Boundary element onto the diagram.

To create a Boundary element as a stereotyped Class, using the Class Properties dialog, follow the steps below

Step	Action
1	Insert a new Class.
2	Right-click on the element and select the Properties context menu option; the Properties dialog displays.
3	In the Stereotype field, type the value boundary .
4	Click on the Apply and OK buttons.
5	Save the diagram (Ctrl+S).

Step	Action

8.8.2 Control

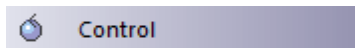


Description

A Control is a stereotyped Object that models a controlling entity or manager. A Control organizes and schedules other activities and elements, typically in Analysis (including Robustness), Sequence and Communication diagrams. It is the controller of the Model-View-Controller pattern.

You can also create a Control as a stereotyped Class.

Toolbox icon



Learn more

- [Create a Control Element](#) ^[1665]
- [Analysis Stereotypes](#) ^[1508]
 - [Sequence Diagram](#) ^[1074]
 - [Communication Diagram](#) ^[1084]
 - [Object Diagram](#) ^[1015]
 - [Analysis Diagram](#) ^[1509]

8.8.2.1 Create a Control Element

How to

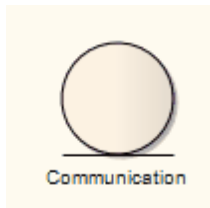
To create a Control element on a diagram as an Object, follow the steps below

Step	Action
1	In the Toolbox, select the More tools Analysis menu option.
2	From the Analysis Elements page, drag the Control element onto the diagram.

To create a Control element as a stereotyped Class, using the Class Properties dialog, follow the steps below

Step	Action
1	Insert a new Class.
2	Right-click on the element and select the Properties context menu option. The Properties dialog displays.
3	In the Stereotype field, type the value control .
4	Click on the Apply and OK buttons.
5	Save the diagram (Ctrl+S).

8.8.3 Entity



Description

An Entity is a stereotyped Object that models a store or persistence mechanism that captures the information or knowledge in a system. It is the Model in the Model-View-Controller pattern.

You can also create an Entity as a stereotyped Class. See the *Create an Entity* topic.

Toolbox icon



Learn more

- [Create an Entity](#) ^[1667]
- [Analysis Stereotypes](#) ^[1508]
- [Sequence Diagram](#) ^[1074]
- [Communication Diagram](#) ^[1084]
- [Object Diagram](#) ^[1015]

- [Analysis Diagram](#) ^[1509]

8.8.3.1 Create an Entity

How to

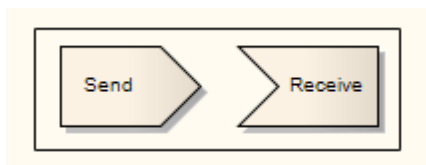
To create an Entity element on a diagram as an Object

Step	Action
1	In the Toolbox, select the More tools Analysis menu option.
2	From the Analysis Elements page, drag the <i>Entity</i> element onto the diagram.

To create an Entity element as a stereotyped Class, using the Class Properties dialog

Step	Action
1	Insert a new Class.
2	Right-click on the element and select the Properties context menu option; the Properties dialog displays.
3	In the Stereotype field, type the value entity .
4	Click on the Apply and OK buttons.
5	Save the diagram (Ctrl+S).

8.8.4 Event



Description

Two elements are used to model *Events*; the:

- *Send Event* which models the generation of a stimulus in the system and the passing of that stimulus

to other elements, either within the system or external to the system

- *Receive Event*, depicted as a rectangle with a recessed 'V' on the left side, which indicates that an event occurs in the system due to some external or internal stimulus; typically this invokes further activities and processing

Send and Receive Events can be added from the Analysis and Activity Element pages of the Toolbox.

If you should select the wrong type of event, or otherwise want to change the type, right-click on the Event and select the **Advanced | Make Sender** or **Advanced | Make Receiver** context menu option, as appropriate.

Toolbox icon



8.8.5 Hyperlinks

Description

You can place a *Hyperlink* element onto a diagram. This element is a type of text element, but one that can contain a pointer to a range of objects such as associated document files, web pages, Help, model features and even other Enterprise Architect model files. When you double-click on the element, Enterprise Architect executes the link.

To add a Hyperlink element, drag the Hyperlink icon from the Common page of the Toolbox onto the diagram.

(Alternatively, click on the **Hyperlink** icon in the UML Elements toolbar and then click on the diagram.)



Configure the Hyperlink:

When you add the Hyperlink to the diagram, the Hyperlink Details dialog displays. If you want to display the information in a more readable layout, you can resize the dialog.

You first select the type of object to link to; click on the drop-down arrow in the **Type** field and select a type. The Hyperlink Details dialog then displays the appropriate fields, prompts or dialog to enable you to specify the object to link to. For example, if you intend to hyperlink to:

- An attribute, the Set Attribute dialog displays to enable you to select that attribute
- A file, the **Action** field displays to enable you to specify whether to Open the file in read only mode, or Edit the file; in either case the file is opened within Enterprise Architect if possible or, if not possible, with the Windows default viewer/editor for the file type

For example, if you hyperlink to a .rtf file, you can *view* it in whichever viewer is appropriate; however, you cannot *edit* .rtf files in Enterprise Architect, so the file always opens in the *Windows* default .rtf editor

- A diagram, the Select a Diagram dialog displays, which enables you to select the diagram from anywhere in the project; you can filter the selection to diagrams of certain types

In most cases you define the location of the hyperlinked object in the **Address** field, either by overtyping the field or by clicking on the (...) (Browse) button. You can use full paths or local (path substitution) paths.

If you select **EA Command** as the link type, the **Address** field is replaced by a drop-down list of Enterprise Architect commands. You can select **LocalPath** and click on the (...) (Browse) button to display the Local Paths dialog, which you complete as required. Subsequently, when you click on the hyperlink the Local Paths dialog immediately displays and you can apply, switch, expand or update the current path.

In the **Alias** field, type the text to display in the hyperlink. If you do not provide an alias, either the text defaults to the link itself, or (for certain link targets such as a matrix profile) the dialog generates a simple text instruction.

If you prefer to display only the hyperlink text, without the icon, select the **Hide Icon** checkbox.

Notes

- If required, you can create a number of empty hyperlinks to complete later; if you then double-click on an empty hyperlink, the Hyperlink Details dialog displays and you can enter the details
- Once you have created the hyperlink, you can also edit the hyperlink text by clicking once on the field and once on the text, then right-clicking and selecting the **Edit Selected** context menu option
- You can add notes to the hyperlink, which display in the Hyperlink Details dialog when you right-click on the hyperlink and select the **Properties** context menu option; you can format these notes using the Notes toolbar

Learn more

- [Hyperlinks to Files](#)^[1669]
- [Hyperlinks to Scripts](#)^[1670]
- [Add Action as a Hyperlink](#)^[1670]
- [Hyperlinks between Diagrams](#)^[1670]

8.8.5.1 Hyperlinks To Files

To create a hyperlink on a diagram to an external file, simply click on the file in a file list (such as Windows Explorer) or on your Desktop and drag it onto the diagram.

A short context menu displays with two options - **Hyperlink** and **Artifact**. Click on the **Hyperlink** option to create the hyperlink on the diagram.

The link is effective immediately, and you can right-click on it to add or change properties as necessary.

Files of most types - including .sql and .ddl - are opened within the appropriate Enterprise Architect code editor.

Learn more

- [Hyperlinks](#)^[1668]

8.8.5.2 Script Hyperlinks

You can create a hyperlink on a diagram to execute a script. Simply drag the required script from the [Execution Analyzer](#)^[1828] window onto the diagram. A context menu displays, from which you select whether the script to be executed is a Build, Test, Run, Debug or Deploy script. The hyperlink is effective immediately; when you click on it, the script executes.

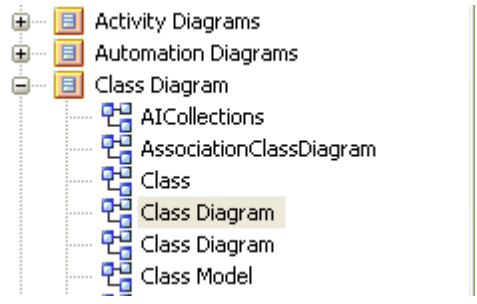
8.8.5.3 Add Action As HyperlinkHow to

To create an Action element to represent a hyperlink, follow the steps below

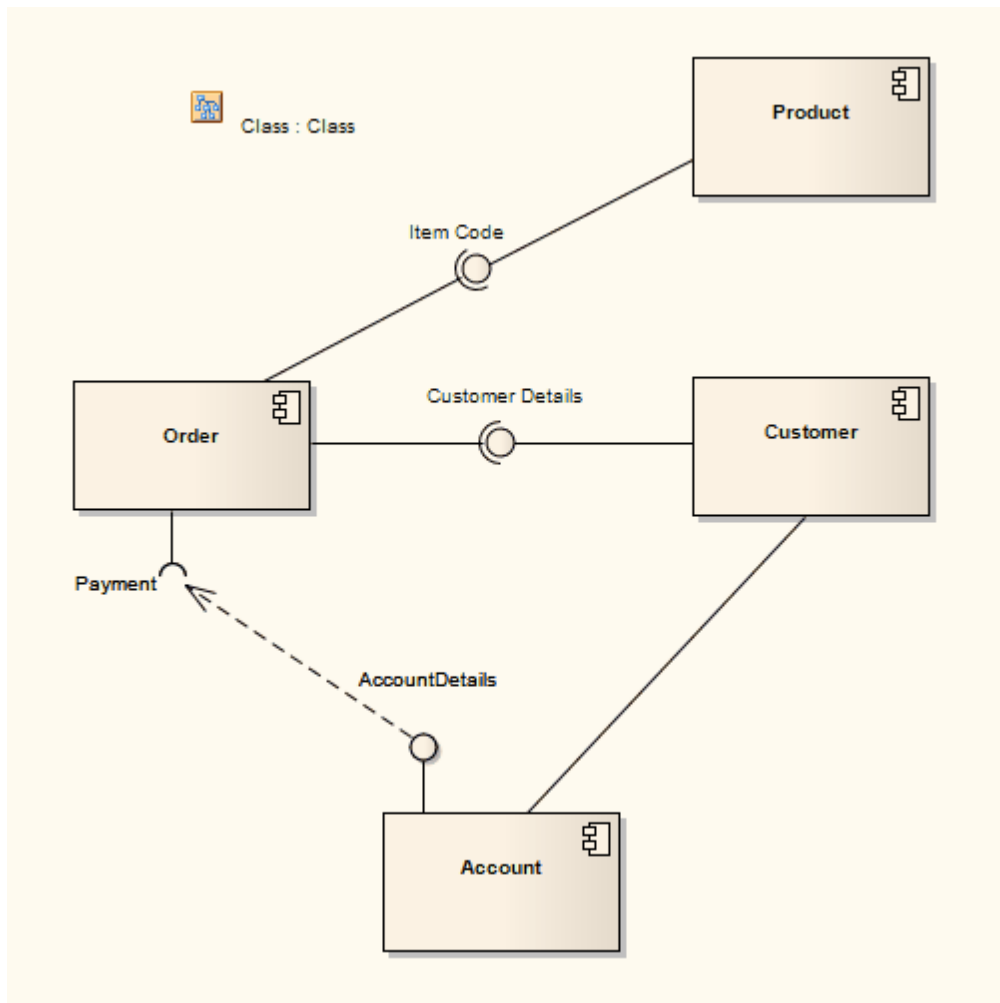
Step	Action
1	Drag an Action element from the Activity page of the Toolbox onto the diagram. A context menu immediately displays.
2	Select the Other menu option. The New Action dialog displays, with the Other radio button selected.
3	Click on the drop down arrow on the field in the Select Kind panel, and click on the Hyperlink option.
4	Click on the OK button. The <i>Hyperlink Action</i> element displays on the diagram.
5	Right click on the element and select the Advanced Set Hyperlink menu option. The Hyperlink Details dialog displays.
6	Set the hyperlink's properties ^[1668] .

8.8.5.4 Hyperlinks Between DiagramsHow to

To create a hyperlink between diagrams

Step	Action
1	<p>Open the diagram in which to display the hyperlink to another diagram.</p> <p>From the Project Browser select the diagram you want to create a hyperlink to:</p>  <p>The screenshot shows a Project Browser tree with the following structure:</p> <ul style="list-style-type: none">Activity DiagramsAutomation DiagramsClass Diagram<ul style="list-style-type: none">AICollectionsAssociationClassDiagramClassClass Diagram (highlighted)Class DiagramClass Model
2	<p>Drag the diagram on to the current diagram.</p> <p>The Select Type dialog displays.</p>
3	<p>Select the Hyperlink option and click on the OK button.</p> <p>The final hyperlinked diagram should resemble the diagram below, where the Class diagram is the diagram to which the Product Order diagram hyperlinks (notice that the hyperlink icon is different).</p>

Example



Notes

- If the hyperlink appears as a Sub Activity, select the **Tools | Options | Diagram | Behavior** menu option and deselect the **Use Automatic SubActivities** checkbox

8.8.6 Image



Description

An *Image* is a System Boundary element that automatically displays first the Boundary Properties dialog and

then the Select Alternate Image dialog to change its representation to an imported image. You can use it as an icon for an element or group of elements, or as a diagram background.

Image elements are available from the Common page of the Toolbox.

Toolbox icon



Learn more

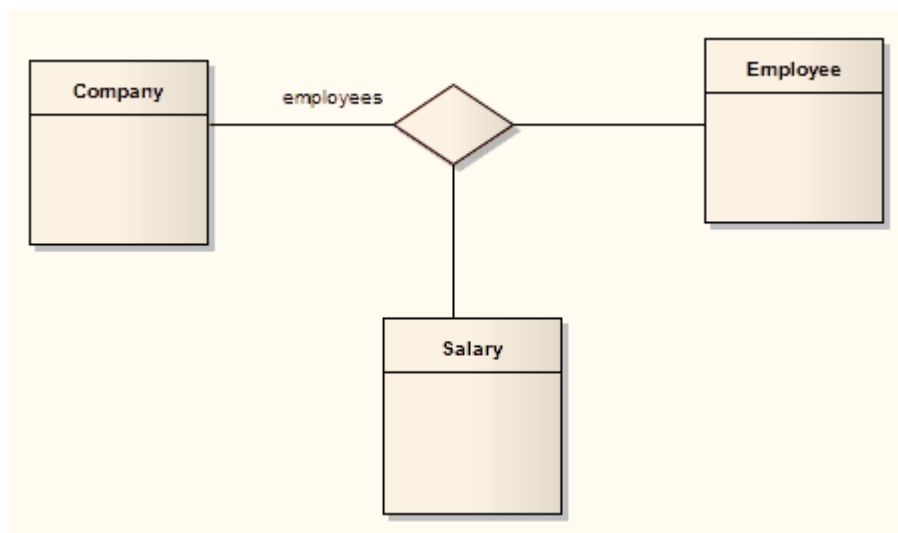
- [System Boundary](#)^[1170]

8.8.7 N-Ary Association



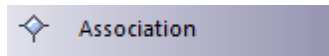
Description

An *n-Ary Association* element is used to model complex relationships between three or more elements, typically in a Class diagram. It is not a commonly-employed device, but can be used to good effect where there is a dependant relationship between several elements. It is generally used with the Association connector, but the relationships can include other types of connector.



In the example above there is a relationship between a *Company*, an *Employee* and a *Salary*.

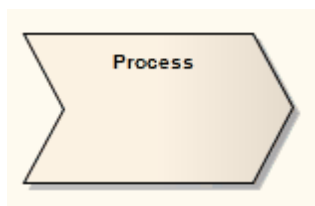
Toolbox icon



Learn more

- [Class Diagram](#) [1014]
- [Association](#) [1210]

8.8.8 Process



Description

A Process is an Activity element with the stereotype *process*, which expresses the concept of a business process. Typically this involves inputs, outputs, work flows, goals and connections with other Processes. The Process element is typically used in Analysis diagrams.

Business processes typically range across many parts of the organization and span one or more systems.

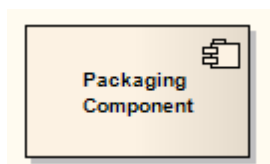
Toolbox icon



Learn more

- [Analysis diagram](#) [1509]

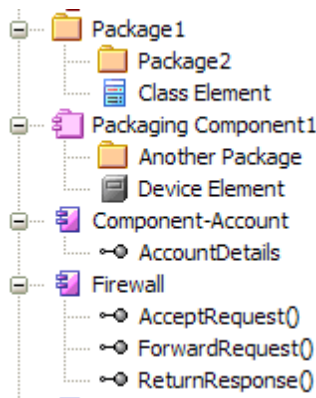
8.8.9 Packaging Component



Description

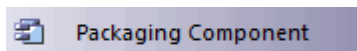
A *Packaging Component* is an element that appears very similar to a Component in a diagram but behaves as a Package in the Project Browser (that is, it can be version controlled and can contain other Packages and elements). It is typically used in Component diagrams.

In the Project Browser, the three elements display as shown below:



The Component element cannot contain child Packages or Packaging Components.

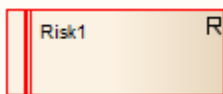
Toolbox icon



Learn more

- [Component diagrams](#)^[1023]
- [Component](#)^[1187]
- [Package](#)^[1200]

8.8.10 Risk



Description

A Risk is defined as the effect of uncertainty on objectives. In Project Management, it is necessary to try to identify risks and assess:

- The likelihood that they have a negative effect on a project and
- How large that effect is likely to be

Those risks with a high probability of occurrence and/or a large impact on the project can be mitigated.

A Risk Management process might consist of the following five steps:

1. Identify risks and represent each with a Risk element.
2. Identify which elements (such as Components, Use Cases or Features) are vulnerable to each risk; you

- might decide to create «*trace*» dependencies from these elements to the Risk elements.
3. Assess the likelihood and magnitude of the risks.
 4. Identify ways to mitigate the risks.
 5. Prioritize the risk reduction measures based on their likelihood, magnitude and ease of mitigation.

Risk *elements* are not the same as the risks that you assign *to* an element through the Project Management window. Such risks are internal to the selected element, whilst a Risk element can be associated with a number of elements, either in a logical group or totally separate.

Risk elements are available from the Requirements page of the Toolbox.

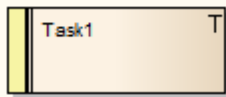
Notes

- Risk elements can be displayed with or without an identifying **R** in the top right corner of the element; to toggle the display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the Options dialog, Objects page

Toolbox icon



8.8.11 Task



Description

A Task element represents a task that must be performed in relation to an element. The Task element enables you to assign resources to the task itself, rather than just to the parent element.

You can create a hierarchy or tree structure of Task elements to break a large task into separate parts and assign different resources to each part.

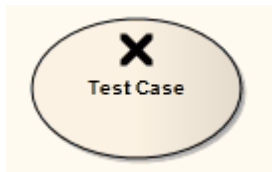
Toolbox icon



Learn more

- [Resource Management](#)^[449]
- [Relationship Matrix](#)^[2247]

8.8.12 Test Case



Description

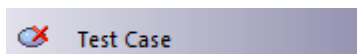
A *Test Case* is a stereotyped Use Case element. You might use it to extend the facilities of the Testing window, by applying element properties and capabilities to the tests of a feature represented by another element or - more appropriately - set of elements. That is, you can define in one go - in the Testing window for the Test Case element - the details of the tests that apply to each of several elements, instead of recording the details separately in each element.

Within the Test Case element properties you can define test requirements and constraints, and associate the test with test files. You can also link the element to Document Artifacts or (in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions) directly to linked documents, such as a Test Plan.

The Test Case element enables you to give greater visibility to tests, in the Project Browser, Diagram List, Package Browser, Model Search, Relationship Matrix, Traceability window and reports.

The Test Case element is available through the Use Case and Maintenance pages of the Toolbox.

Toolbox icon



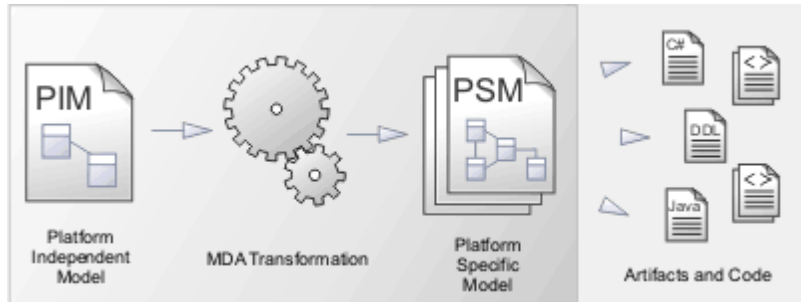
Learn more

- [The Testing Workspace](#) 

Part

IX

9 Model Transformation



Model Driven Architecture (MDA) Transformations provide a fully configurable way of converting model elements and model fragments from one domain to another. This typically involves converting Platform-Independent Model (PIM) elements to Platform-Specific Model (PSM) elements. A single element from the PIM can be responsible for creating multiple PSM elements across multiple domains.

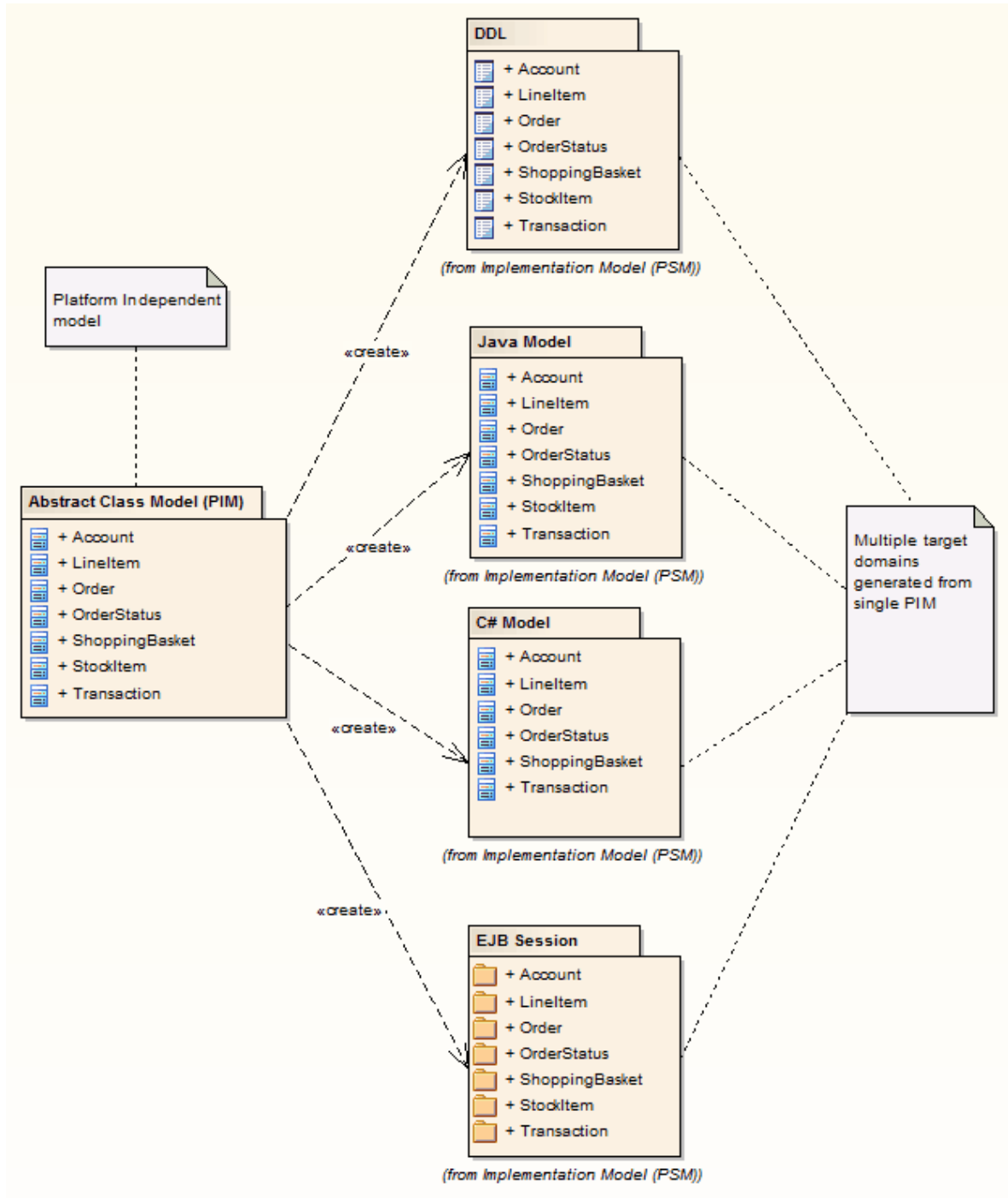
Topics

Topic	Detail	See also
Overview	<p>Transformations are a huge productivity boost, and reduce the necessity of manually implementing stock Classes and elements for a particular implementation domain; for example, database tables generated from persistent PIM Classes.</p> <p>Enterprise Architect includes a number of basic built-in Transformations (see <i>Included Transformations</i>, below); Sparx Systems intend to make further Transformations available over time, either as built in Transformations or as downloadable modules from the Sparx Systems website.</p> <p>For a further productivity boost, Enterprise Architect can automatically generate code for your transformed Classes that target code languages; see the Generate Code on result option on the Model Transformation dialog.</p> <p>A Transformation is defined using Enterprise Architect's simple code generation template language, and involves no more than writing a template to create a simple intermediary source file; Enterprise Architect reads the source file and binds that to the new PSM.</p>	Transform Elements <small>[1683]</small>
Transformation Dependencies	<p>Enterprise Architect also creates internal bindings (Transformation Dependencies) between each PSM created and the original PIM.</p> <p>This is essential, as it enables you to forward synchronize from the PIM to the PSM many times, adding or deleting features as you go; for example, adding a new attribute to a PIM Class can be forward synchronized to a new column in the Data Model.</p> <p>You can observe the Transformation Dependencies for a package using the Traceability window; this enables you to check the impact of changes to a PIM element on the corresponding elements in each generated PSM, or to verify where a change required in a PSM should be initiated in the PIM</p>	Traceability <small>[635]</small>

Topic	Detail	See also
	<p>(and also to reflect back in other PSMs).</p> <p>The Transformation Dependencies are a valuable tool in managing the traceability of your models.</p> <p>Enterprise Architect does not delete or overwrite any element features that were not originally generated by the transform; therefore, you can add new methods to your elements, and Enterprise Architect does not act on them during the forward generation process.</p>	
Included Transformations	<p>Transformations that are currently built-in include:</p> <ul style="list-style-type: none"> • C# - Converts a PIM to a standard C# implementation set • Data Model to ERD - Transforms a Data Model to an Entity Relationship Diagram (ERD) • DDL - Transforms platform-independent Class elements to platform-specific table elements • EJB Entity - Transforms platform-independent Class elements to packages containing the Class and Interface elements that comprise an EJB Entity Bean • EJB Session - Transforms platform-independent Class elements to packages containing the Class and Interface elements that comprise an EJB Session Bean • ERD to Data Model - Transforms an Entity Relationship Diagram into a Data Model • Java - Transforms platform-independent elements to Java language elements • JUnit - Converts a Java model to a model where test methods are created for each public method of any original Class • NUnit - Converts a .Net language specific model to a model where test methods are created for each public method of any original Class • WSDL - Converts a simple representation of a WSDL interface into the elements required to generate that interface • XSD - Transforms platform-independent elements to XSD elements 	<p>Built-in Transformations ^[1686]</p> <p>C# Transformation ^[1687]</p> <p>Data Model To ERD Transformation ^[1689]</p> <p>DDL Transformation ^[1689]</p> <p>EJB Transformations ^[1694]</p> <p>EJB Transformations ^[1694]</p> <p>ERD To Data Model Transformation ^[1697]</p> <p>Java Transformation ^[1699]</p> <p>JUnit Transformation ^[1702]</p> <p>NUnit Transformation ^[1704]</p> <p>WSDL Transformation ^[1705]</p> <p>XSD Transformation ^[1707]</p>

Example

The following diagram highlights how Transformations work and how they can significantly boost your productivity:



Notes

- If you are using the Corporate, Business and Software Engineering, System Engineering or Ultimate edition, if security is enabled you must have **Transform Package** access permission to perform an

MDA Transform on a package

Learn more

- [Permission List](#)^[264]
- [Import Code Generation and Transformation Templates](#)^[1401]
- [Export Code Generation and Transformation Templates](#)^[1400]
- [Transformation Templates](#)^[1710]
- [Write Transformations](#)^[1712]
- [Chaining Transformations](#)^[1684]

Learning Center topics

- **(Alt + F1) | Model Transformations**

9.1 Transform Elements

There are two modes for initiating a Model Transformation, each of which can be started in two ways.

Access

To transform selected elements on a diagram: **Tools | Model Transformation (MDA) | Transform Selected Elements**, or

Class context menu | Transform (Ctrl + H)

To transform elements in the package currently selected in the Project Browser:

Package, or **Tools | Model Transformation (MDA) | Transform Current Package**, or **Project Browser package context menu | Transform**

Use to

- Transform elements
- Transform packages

Reference

Field	Usage	See also
Elements	Indicates the individual elements to be included in the transformation.	
All	Indicates all of the elements from the list to be included in the transformation.	
None	Deselects all of the elements from the list.	
Include child packages	Specifies inclusion of elements in child packages of the selected package.	
Transformations	Indicates which transformations to perform and the package each of them should be transformed to (use the (...) button to select the package in which the transformed elements are being created).	
Generate Code on result	Specify whether or not to automatically generate code for transformed Classes that target code languages. Automatically generating code helps boost productivity in development. With this option selected, the first time you transform to the selected Class Enterprise Architect enables you to select a	

Field	Usage	See also
	filename to generate to; subsequent transformations automatically generate any Class with a filename set.	
Perform Transformations on result	Specify if transformations previously done on target Classes should be automatically executed.	Chaining Transformation s ^[1684]
Intermediary File	Specify the filename of the intermediary file (if any).	
Write Always	Write the intermediary file to disk.	Intermediary Language ^[1714]
Write Now	Generate the intermediary file but do not perform the transform.	Intermediary Language Debugging ^[1714]
Do Transform	Execute the transform command.	

Notes

- When the dialog displays, all elements are selected and all transformations previously performed from any of these Classes are checked

Learning Center topics

- **(Alt + F1) | Model Transformations | Transform | Transform Elements**
- **(Alt + F1) | Model Transformations | Transform | Transform Packages**

9.1.1 Chaining Transformations

Chaining transformations provides an extra degree of flexibility and power to transformations.

Topics

Topic	Detail	See also
Example	Two transformations have a common element; you might separate this element out into its own transformation, and then perform the original transformations from the common point. The separated transform could even produce a useful model itself.	
Setting	Enterprise Architect provides for chaining transformations, by enabling you to set transformations that have already been performed on target Classes to be performed automatically next time that Class is transformed to. To do this, select the Perform Transformations on result checkbox	

Topic	Detail	See also
	in the Model Transformation dialog.	

9.2 Built-in Transformations

Enterprise Architect provides a number of built-in transformations. These transformations are intended to be useful to as many users as possible, acting as a good base to modify to include the specifics of your custom domain, and to be good examples of how to write transformations.

The following transformations are included in Enterprise Architect:

Transformations	Detail	See also
C#	Converts Platform-Independent Model (PIM) elements to language-specific C# Class elements.	C# Transformation 1687
Data Definition Language	Creates a data model from the logical model, generating a model targeted at the default database type that is ready for DDL generation.	DDL Transformation 1689
Entity Relationship Diagram to Data Model	Creates a data model from the ERD logical model, generating a model targeted at the default database type ready for DDL generation.	ERD To Data Model Transformation 1697
Data Model to Entity Relationship Diagram	A reverse of the ERD to Data Model transformation.	Data Model To ERD Transformation 1689
EJB Session Bean	Generates a single Class elements to EJB session.	EJB Transformations 1694
EJB Entity Bean	Generates a single Class elements to EJB entity.	EJB Transformations 1694
Java	Converts Platform-Independent Model (PIM) elements to language-specific Java Class elements.	Java Transformation 1699
JUnit	Creates a Class with test methods for all public methods of an existing Java Class.	JUnit Transformation 1702
NUnit	Creates a Class with test methods for all public methods of an existing .Net compatible Class.	NUnit Transformation 1704
WSDL	Creates an expanded model of a WSDL interface from a simple model, which is suitable for generation.	WSDL Transformation 1705
XSD	Converts Platform-Independent Model (PIM) elements to UML Profile for XML elements as an intermediary step to creating an XML Schema.	XSD Transformation 1707

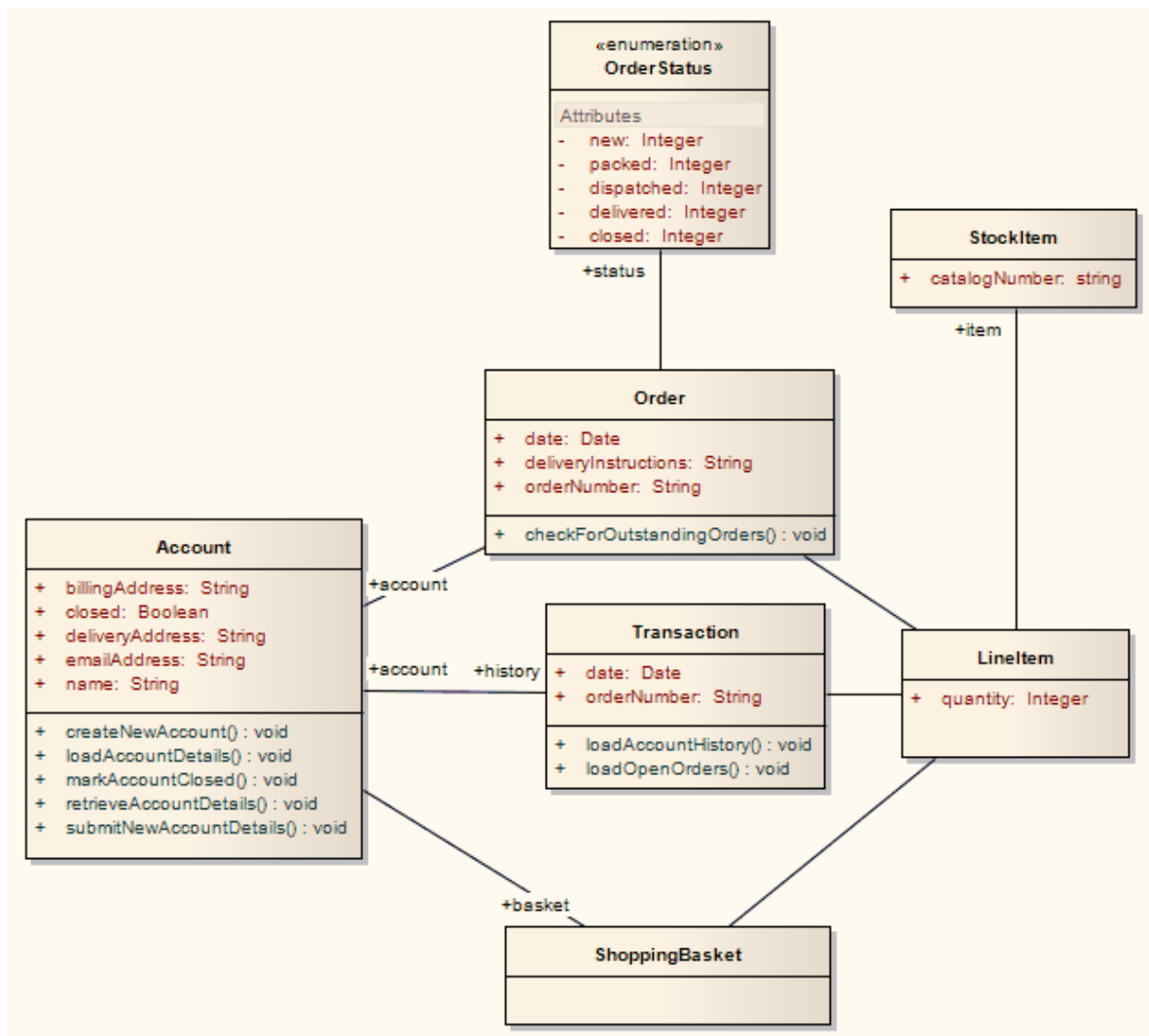
9.2.1 C# Transformation

The **C# transformation** converts Platform-Independent Model (PIM) elements to language-specific C# Class elements.

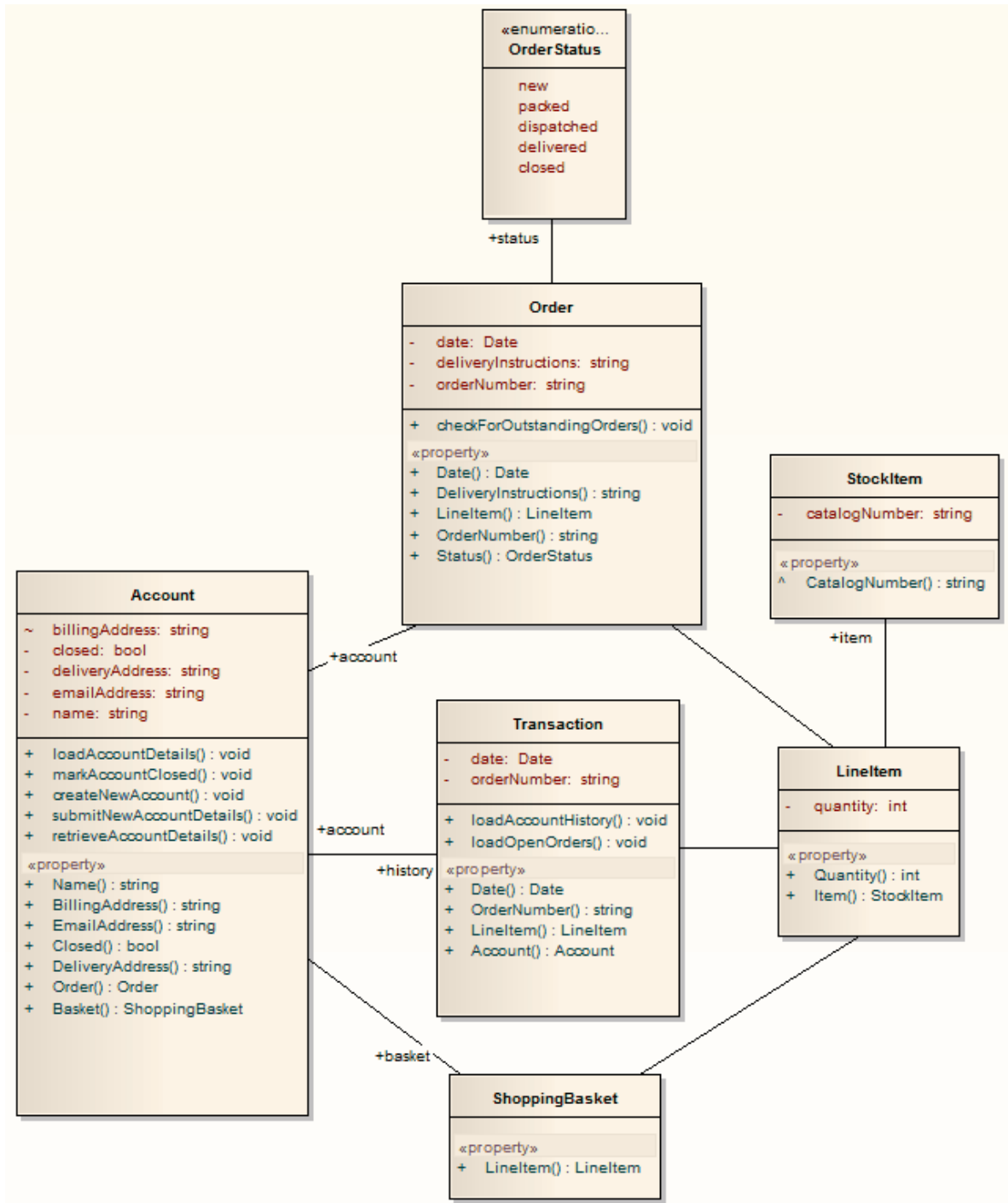
The transformation converts PIM model types to C# types and creates encapsulation according to Enterprise Architect's options for creating properties from C# attributes, which you set on the C# Specifications page of the Options dialog.

Example

The Platform-Independent Model (PIM):



After transformation, becomes the PSM:



[Learn more](#)

- [C# Options Page](#) ¹⁹¹²

9.2.2 Data Model To ERD Transformation

The **Data Model to ERD transformation** is a reverse engineering of the **ERD to Data Model transformation**.

The transformation uses and demonstrates support in the intermediary language for the following database-specific concepts:

Reference

Concept	Definition	See also
Entity	Mapped one-to-one onto table elements.	
Attribute	Mapped one-to-one onto columns.	
Primary Key	Comes from the PrimaryKey type of column.	

Notes

- Sometimes you might want to limit the stretch of the diamond-shape Relationship connectors; simply pick a Relationship connector, right-click to display the context menu, and select the **Bend Line at Cursor** option

Learn more

- [ERD to Data Model Transformation](#) ^[1697]
- [Data Models](#) ^[1625]

9.2.3 DDL Transformation

The purpose of the **DDL transformation** is to create a data model from the logical model, generating a model targeted at the default database type that is ready for DDL generation. The data model can then be used to automatically generate DDL statements to run in one of the Enterprise Architect supported database products.

The **DDL Transformation** uses and demonstrates support in the **intermediary language** for the following database-specific concepts:

Reference

Concept	Definition	See also
Table	Mapped one-to-one onto Class elements. 'Many to Many' relationships are supported by the transformation creating Join tables.	

Concept	Definition	See also
Column	Mapped one-to-one onto attributes.	
Primary Key	Lists all the columns involved; this ensures that they exist in the Class and creates a primary key method for them.	
Foreign Key	This is a special sort of connector. The <i>Source</i> and <i>Target</i> sections list all of the columns involved; this ensures that they exist and that a matching primary key exists in the destination Class, and that the transformation creates the appropriate foreign key.	Transform Foreign Keys <small>[1728]</small>

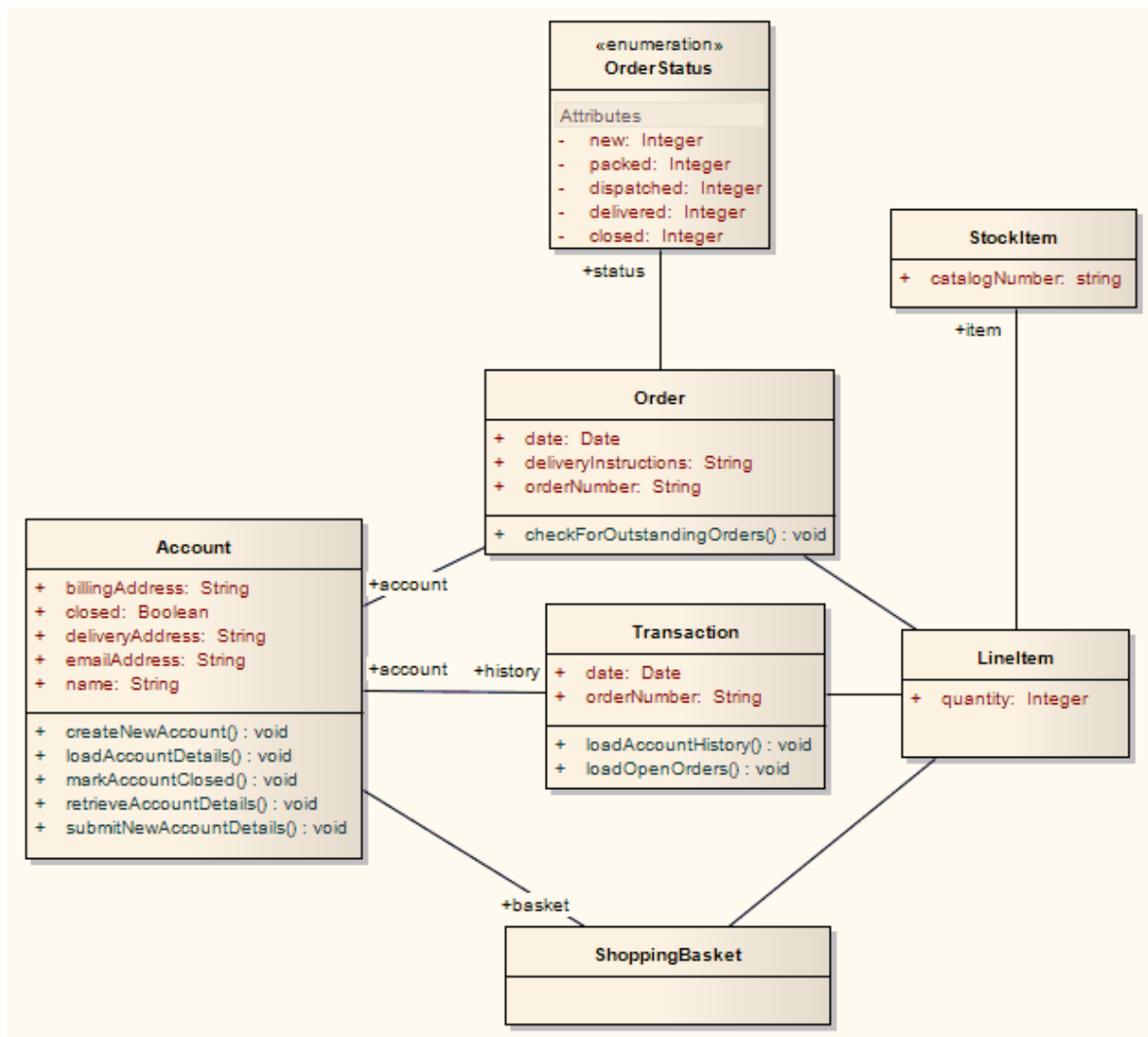
Notes

- DBMS specific aspects not depicted in a Logical model such as Stored Procedures, Triggers, Views and Check Constraints are definable post transformation; see *Physical Data Model* for details

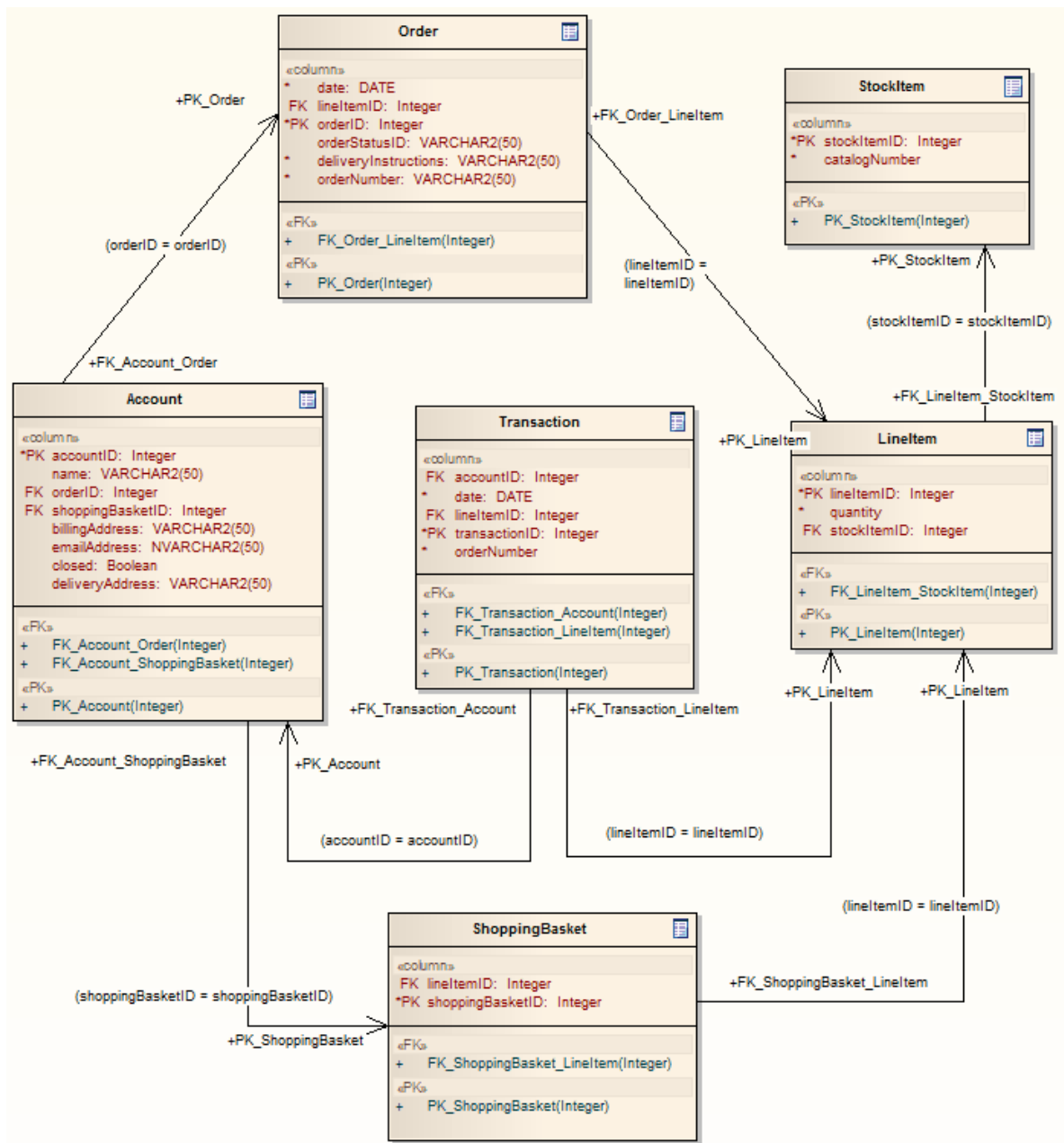
Example

The following two diagrams show a typical PIM to Data Model Transformation.

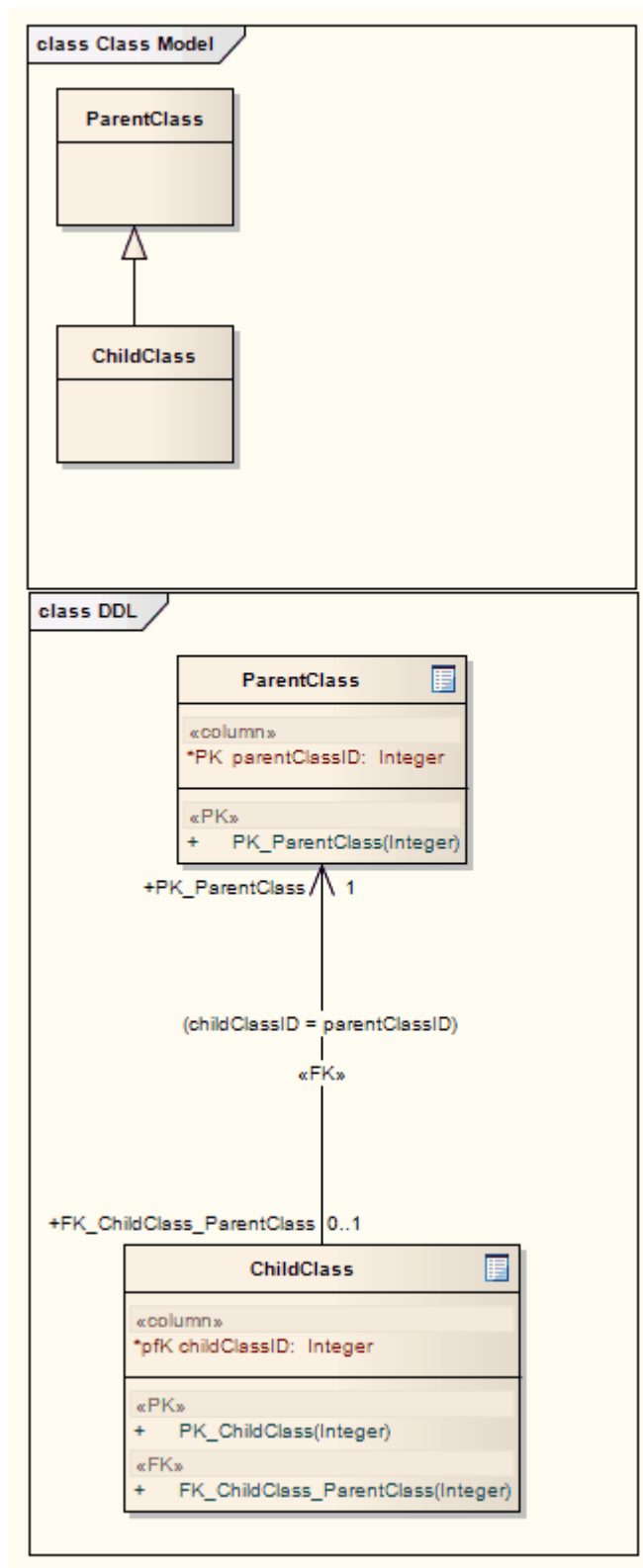
The Platform-Independent Model (PIM):



After transformation becomes the PSM:



Generalizations are handled by providing the child element with a foreign key to the parent element, as in the following diagram. Copy-down inheritance is not supported.



Learn more

- [Intermediary Language](#) ^[1714]
- [Data Models](#) ^[1625]

- [Physical Data Model](#)^[1932]

9.2.4 EJB Transformations

The purpose of the **EJB Session Bean transformation** and the **EJB Entity Bean transformation** is to reduce the work required in generating the internals of Enterprise Java Beans, thus enabling you to concentrate on modeling at a higher level of abstraction.

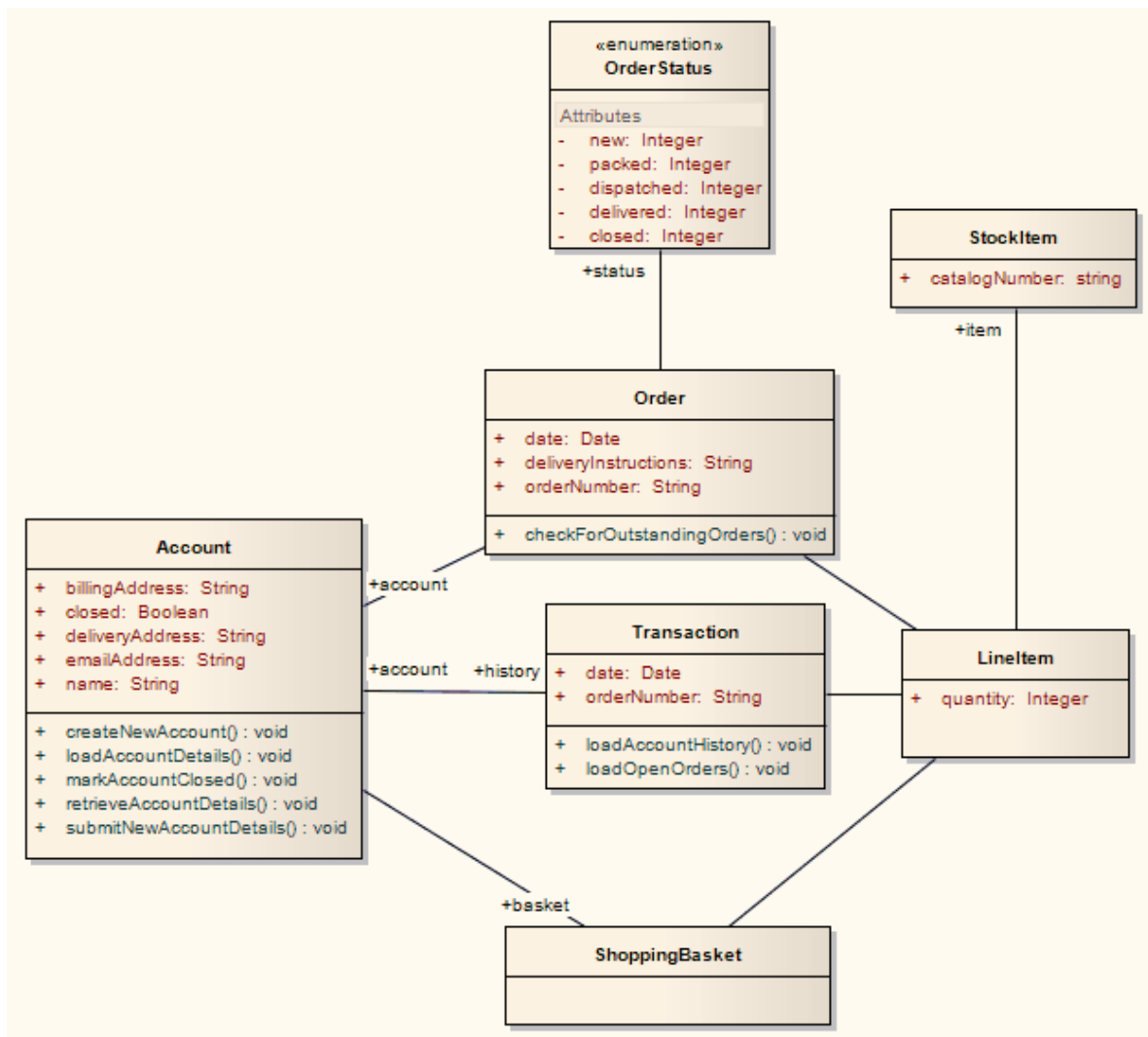
Topics

Topic	Detail	See also
EJB Session Bean transformation	<p>The EJB Session Bean transformation generates the following from a single Class element containing the attributes, operations and references required for code generation by the javax.ejb.* package:</p> <ul style="list-style-type: none"> • An implementation Class element • A home interface element • A remote interface element 	
EJB Entity Bean transformation	<p>The EJB Entity Bean transformation generates the following from a single Class element containing the attributes, operations and references required for code generation by the javax.ejb.* package:</p> <ul style="list-style-type: none"> • An implementation Class element • A home interface element • A remote interface element • A primary key element 	

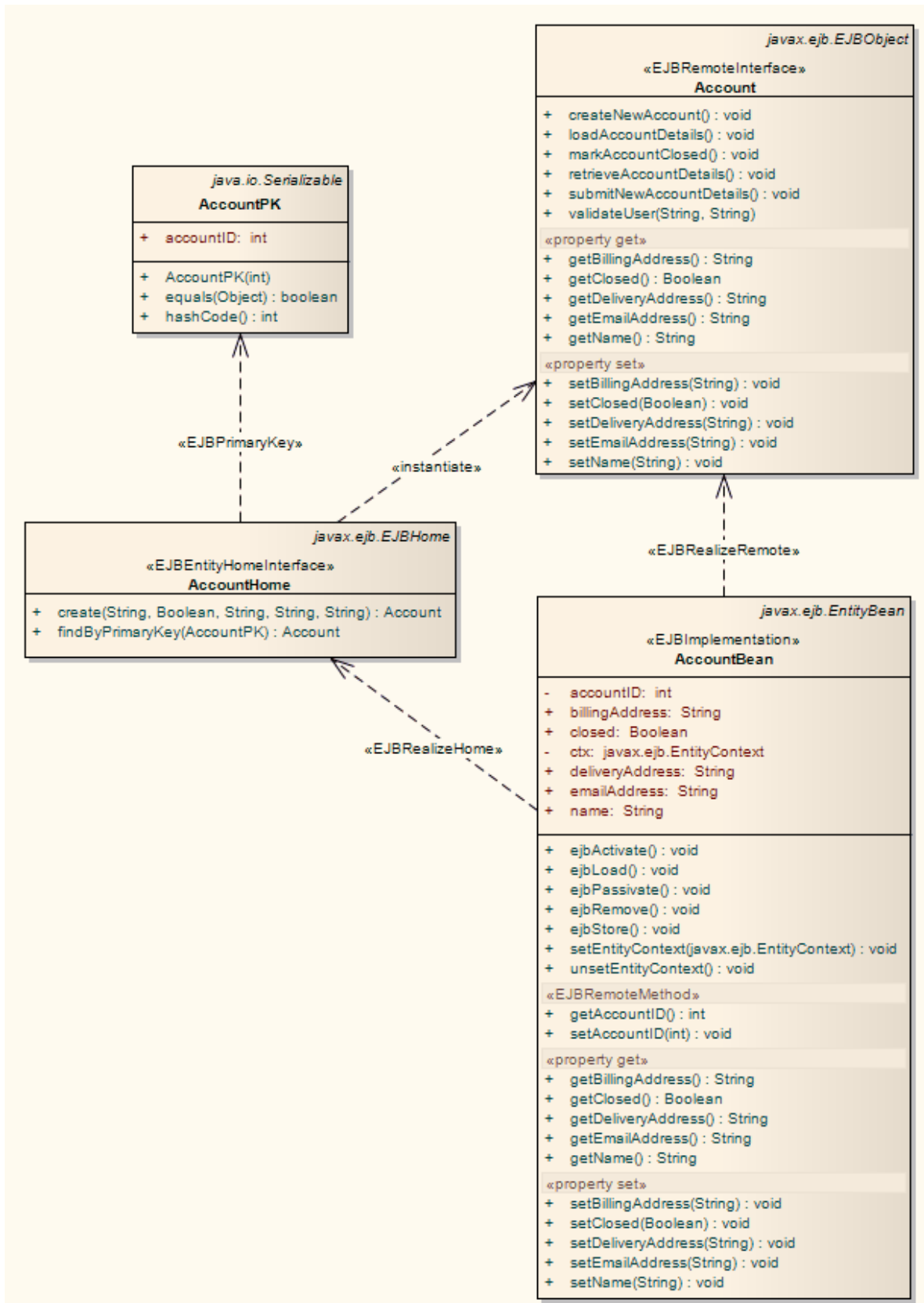
Both transformations also generate a META-INF package containing a deployment descriptor element.

Example

The Platform-Independent Model (PIM):



After transformation generates a set of Entity Beans, where each one takes the following form (for the Account Class):



Learn more

- [Software Engineering](#)^[1733]
- [Generate Source Code](#)^[1768]

9.2.5 ERD To Data Model Transformation

The purpose of the **Entity Relationship Diagram (ERD) to Data Model transformation** is to create a data model from the ERD logical model, generating a model targeted at the default database type ready for DDL generation. Before doing the transformation, make sure you have defined the common data type for each attribute and selected a database type as the default database. The data modeling diagram can then be automatically generated. This data model can be used for generating DDL statements to run in one of the Enterprise Architect supported database products.

The transformation uses and demonstrates support in the intermediary language for the following database-specific concepts:

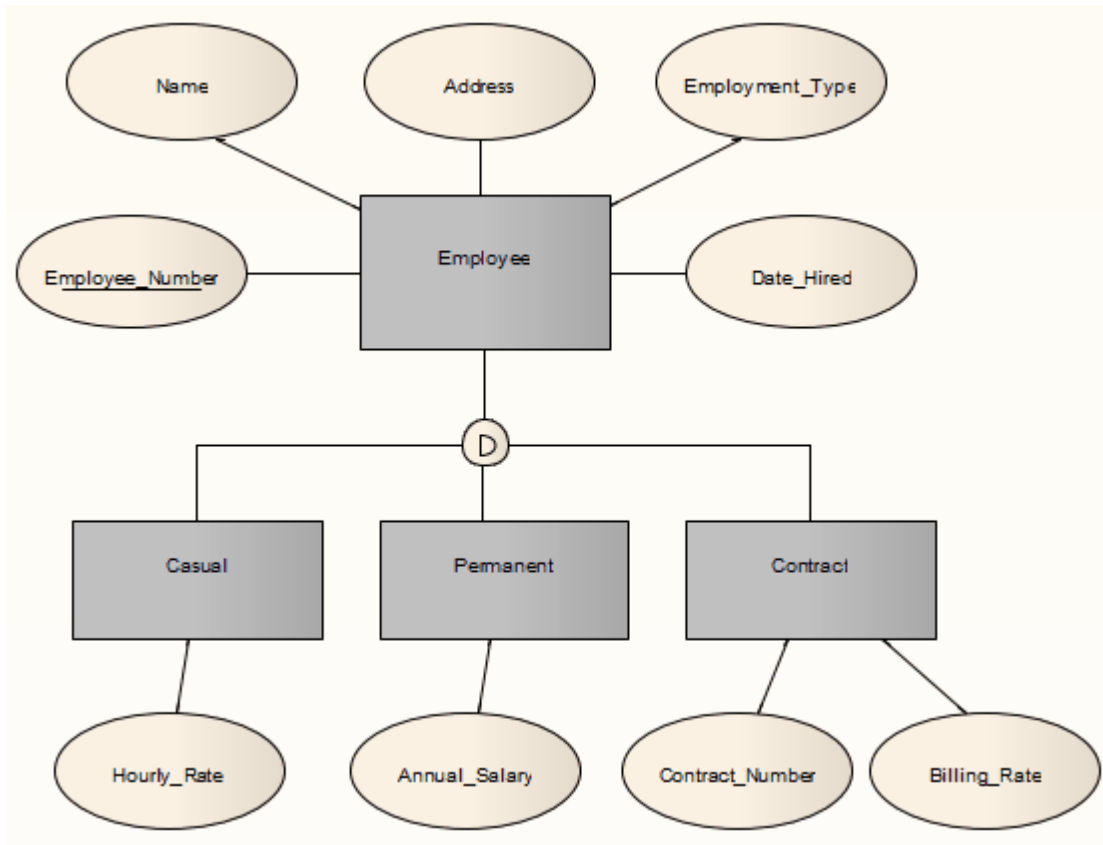
Reference

Concept	Definition	See also
Table	Mapped one-to-one onto Class elements.	
Column	Mapped one-to-one onto attributes.	
Primary Key	Lists all the columns involved; this ensures that they exist in the Class and creates a primary key method for them.	
Foreign Key	This is a special sort of connector. The <i>Source</i> and <i>Target</i> sections list all of the columns involved; this ensures that they exist and that a matching primary key exists in the destination Class, and that the transformation creates the appropriate foreign key.	

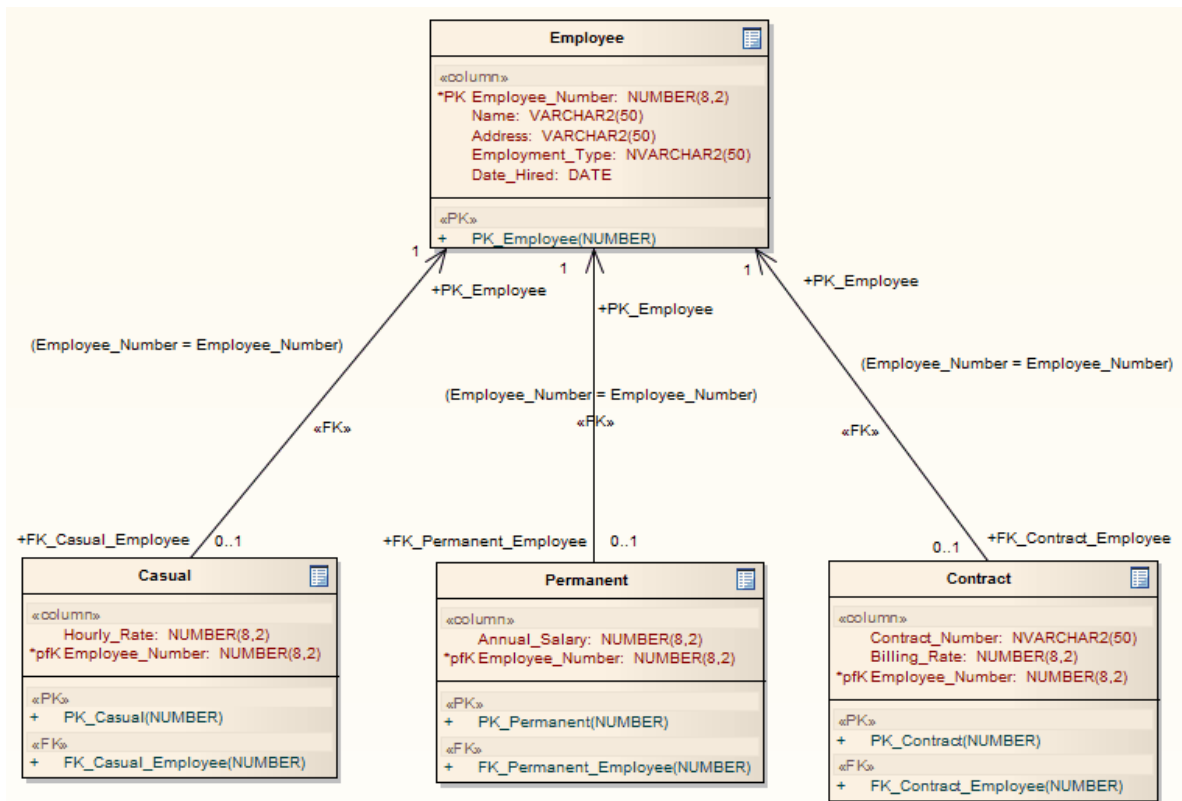
Generalization

Generalization can be handled in ERD technology, as illustrated by the following example. Note that the copy-down inheritance is currently supported with two levels only.

Example



This transforms to:



Notes

- Sometimes you might go back to the ERD, make some changes and then want to do another transformation. To achieve better results, always delete the previous transformation package before doing the next transformation

Learn more

- [Intermediary Language](#) [1714]
- [Data Models](#) [1625]
- [Physical Data Model](#) [1932]

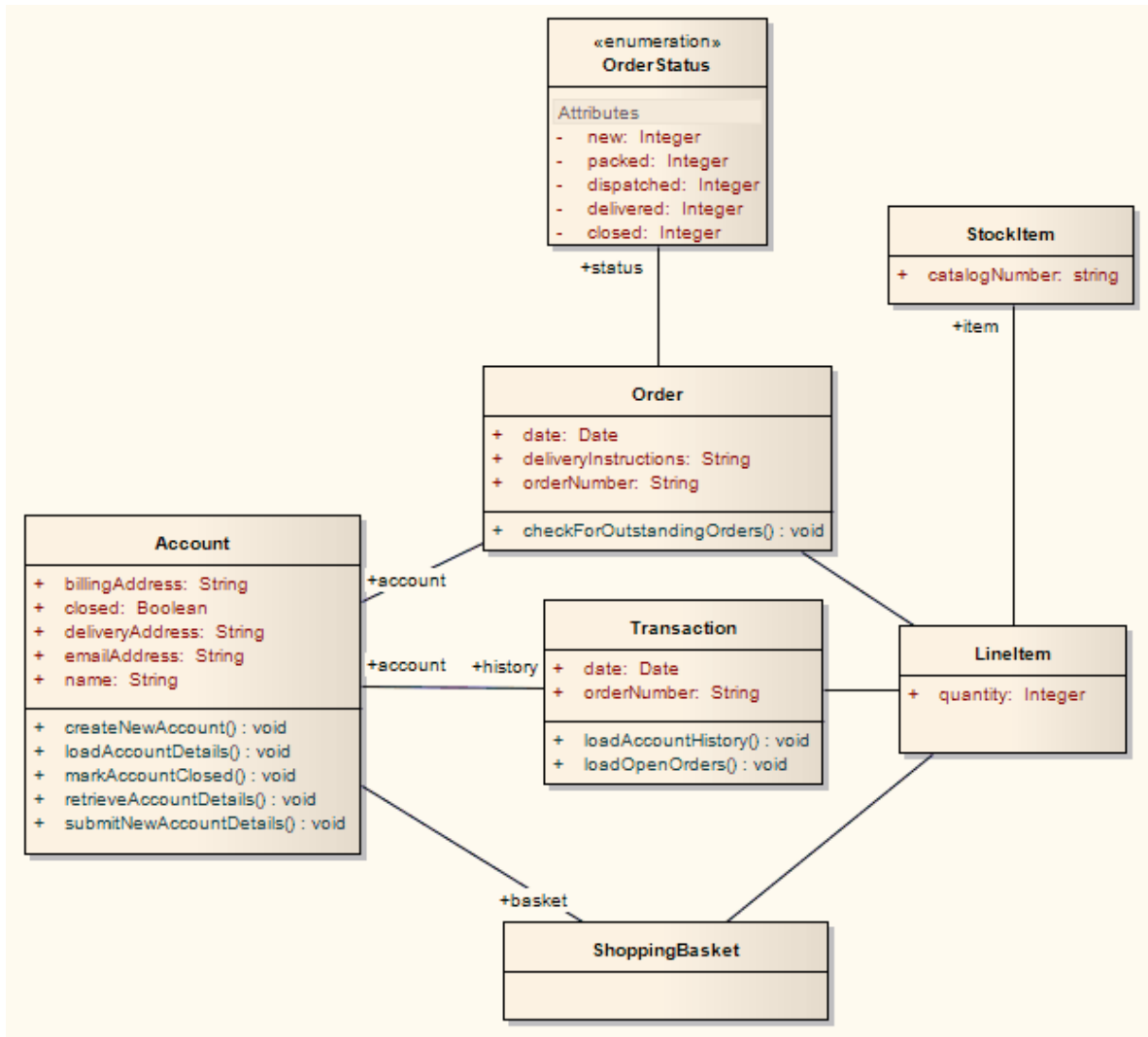
9.2.6 Java Transformation

The purpose of the **Java transformation** is to convert Platform-Independent Model (PIM) elements to language-specific Java Class elements. The transformation converts the PIM model types to Java types and creates encapsulation according to Enterprise Architect's options for creating properties from Java attributes; that is, producing the getters and setters according to the rules you have defined. Notice that the *public* attributes in the PIM are converted to *private* attributes in the PSM.

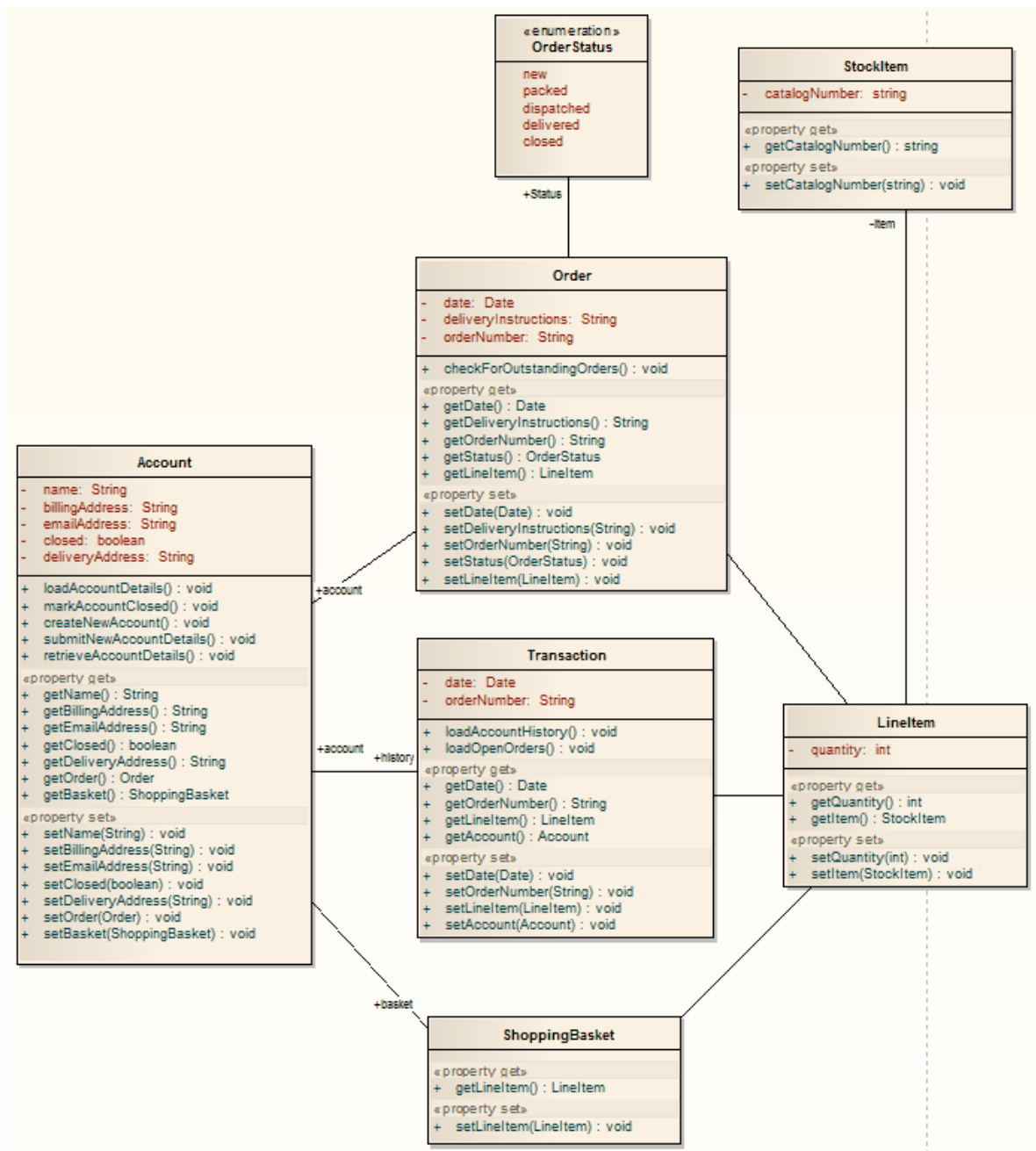
You set the code generation options for Java code generation on the **Java Specifications** page of the **Options** dialog.

Example

The Platform-Independent Model (PIM):



After transformation becomes the PSM:



Learn more

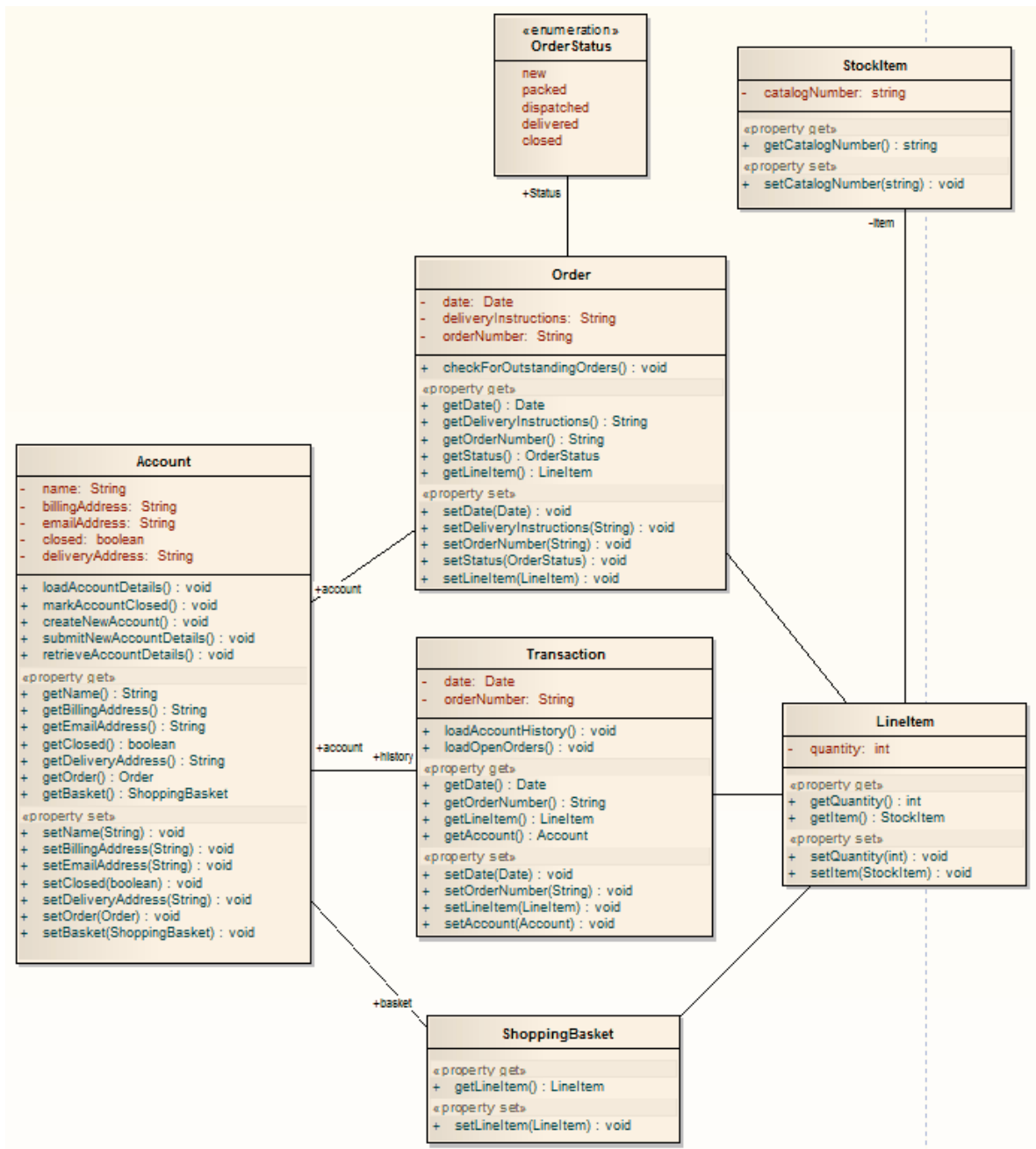
- [Java Options](#)¹⁹¹⁵
- [Software Engineering](#)¹⁷³³
- [Generate Source Code](#)¹⁷⁶⁸

9.2.7 JUnit Transformation

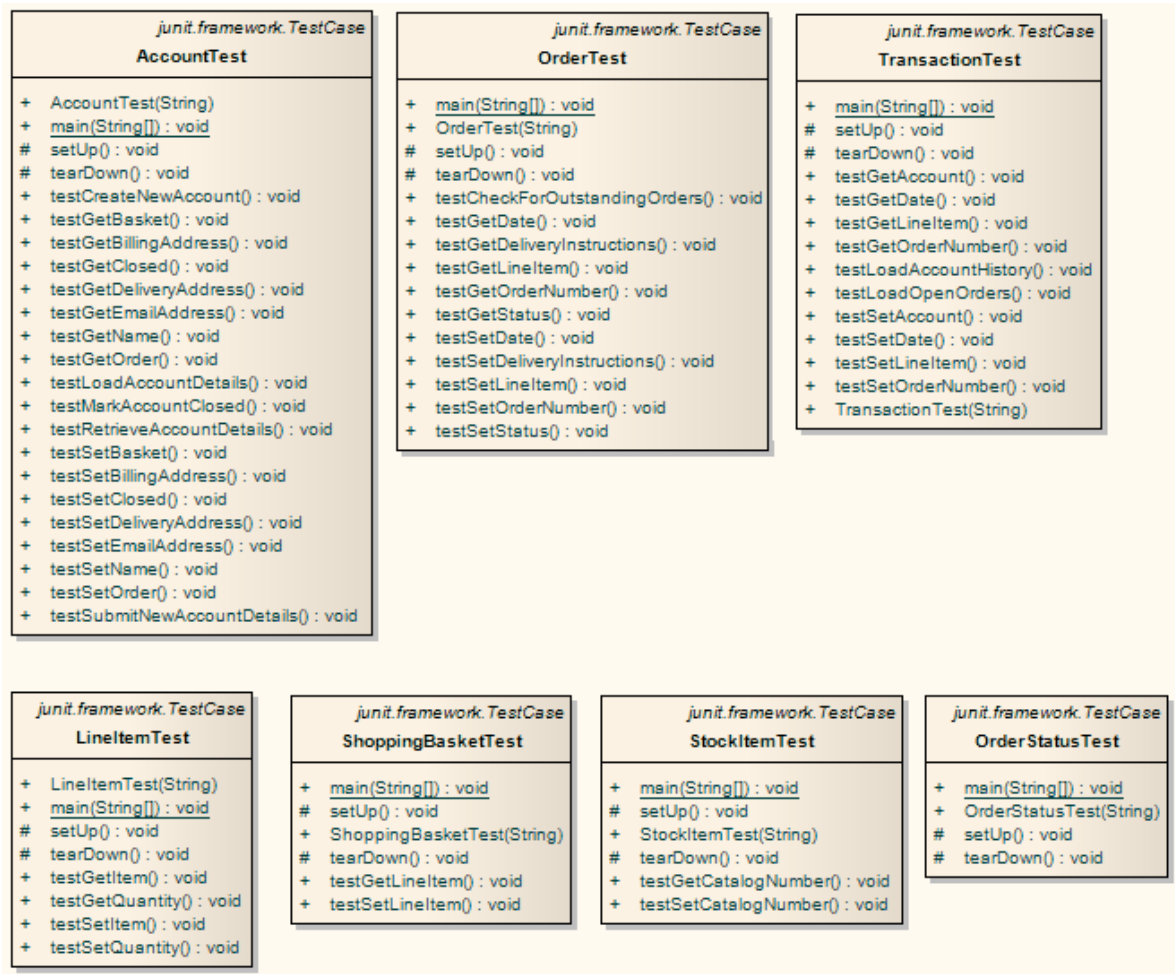
The purpose of the **JUnit transformation** is to create a Class with test methods for all public methods of an existing Java Class. The resulting Class can then be generated and the tests filled out and run by JUnit.

Example

The Java model originally transformed from the PIM:



After transformation becomes the PSM:



Notes

- Note that for each Class in the Java model, a corresponding test Class has been created. Each of these test Classes contains a test method for every public method in the source Class, plus the methods required to appropriately set up the tests. It is your responsibility to fill in the details of each test

Learn more

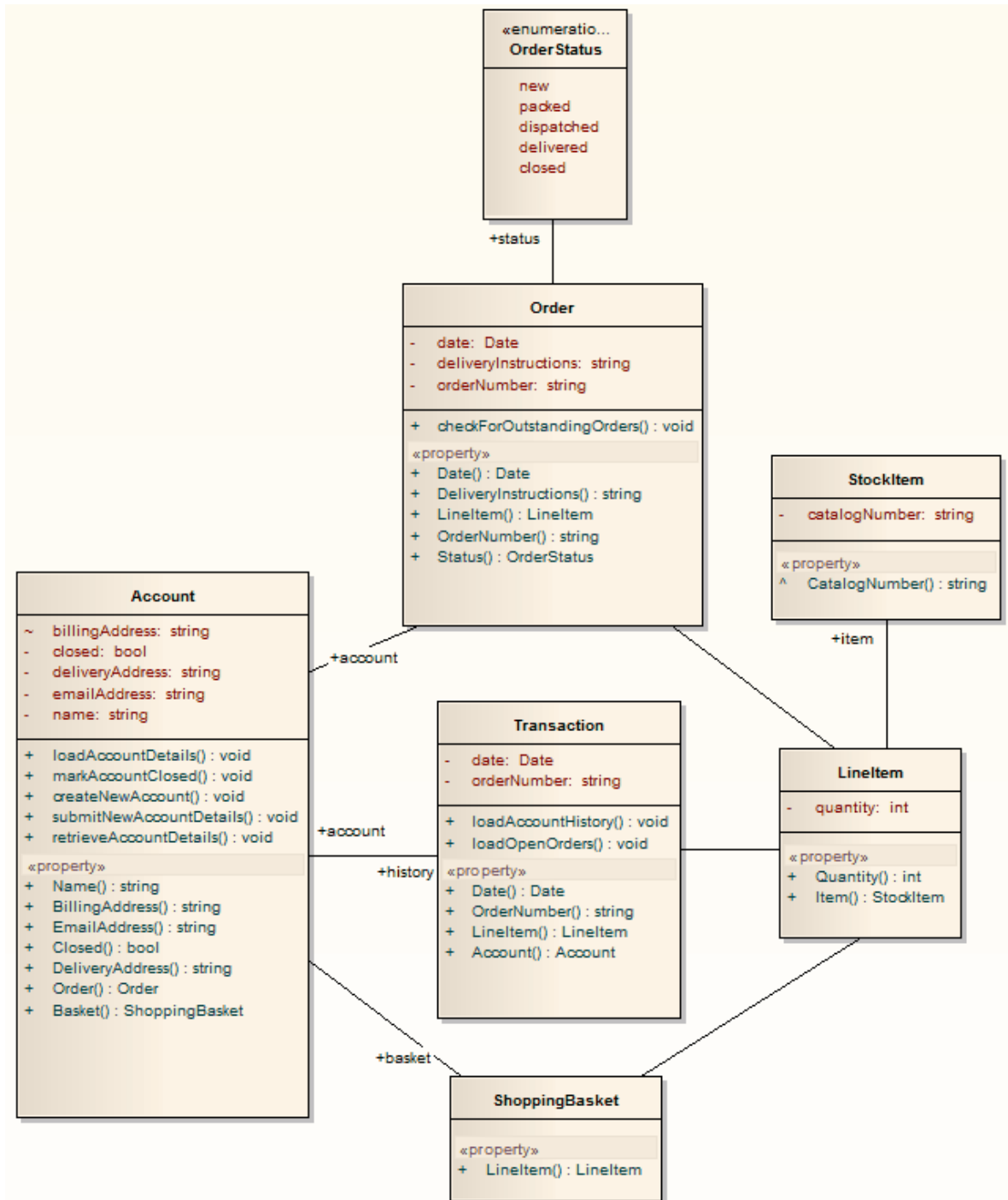
- [Unit Testing](#) ^[2183]

9.2.8 NUnit Transformation

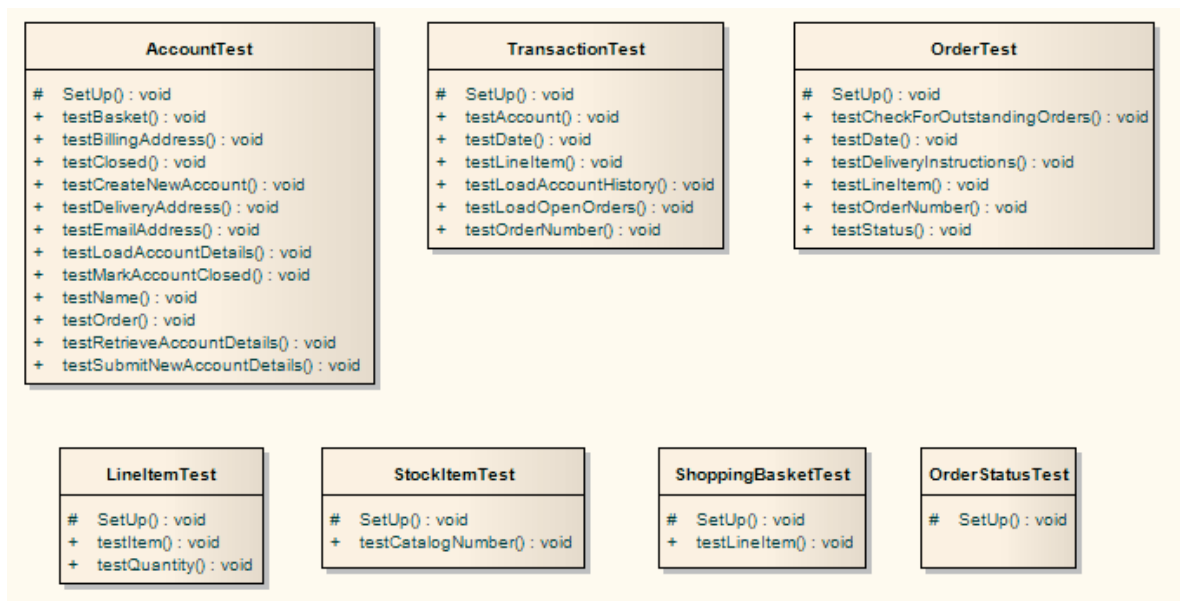
The purpose of the **NUnit transformation** is to create a Class with test methods for all public methods of an existing .Net compatible Class. The resulting Class can then be generated and the tests filled out and run by NUnit.

Example

The C# model originally transformed from the PIM:



After transformation becomes the PSM:



Notes

- Note that for each Class in the C# model, a corresponding test Class has been created. Each of these test Classes contains a test method for every public method in the source Class, plus the methods required to appropriately set up the tests. It is your responsibility to fill in the details of each test

Learn more

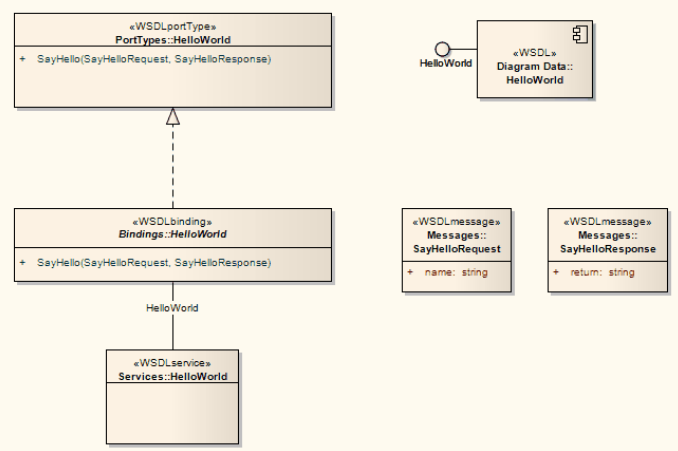
- [Unit Testing](#)^[2183]

9.2.9 WSDL Transformation

The purpose of the **WSDL transformation** is to create from a simple model an expanded model of a WSDL interface that is suitable for generation.

Topics

Topic	Detail	See also
WSDL Transformation	<p>Take the following example interface:</p> <pre> classDiagram class HelloWorld { <<interface>> + SayHello(name :string) : string } </pre> <p>This generates the corresponding WSDL component, service,</p>	<p>XSD Transformation^[1707]</p> <p>WSDL Generation^[2085]</p>

Topic	Detail	See also
	<p>port type, binding and messages as follows.</p> <ul style="list-style-type: none"> • Classes are handled in the same way as the XSD Transformation • All <i>in</i> parameters are transformed into WSDL <i>Message Parts</i> in the <i>Request</i> message • The return value and all <i>out</i> and <i>return</i> parameters are transformed into WSDL <i>Message Parts</i> in the <i>Response</i> message • All methods where a value is returned are transformed into <i>Request-Response</i> operations while all methods not returning a value are transformed into <i>OneWay</i> operations • The transformation does not handle generation of <i>Solicit-Response</i> and <i>Notification</i> methods or faults  <p>The diagram illustrates the WSDL components for a service named 'HelloWorld'. It features a package structure with the following elements:</p> <ul style="list-style-type: none"> PortType: «WSDLportType» PortTypes::HelloWorld, containing the operation + SayHello(SayHelloRequest, SayHelloResponse). Binding: «WSDLbinding» Bindings::HelloWorld, containing the operation + SayHello(SayHelloRequest, SayHelloResponse). It is associated with the PortType via a dashed line labeled 'HelloWorld'. Service: «WSDLservice» Services::HelloWorld, associated with the Binding via a solid line labeled 'HelloWorld'. Messages: Two «WSDLmessage» elements: Messages::SayHelloRequest (with + name: string) and Messages::SayHelloResponse (with + return: string). Diagram Data: «WSDL» Diagram Data::HelloWorld, associated with the PortType via a solid line labeled 'HelloWorld'. <p>The resulting package can then have the specifics filled out using the WSDL editing capabilities of Enterprise Architect, and finally be generated using WSDL generation.</p>	

Learn more

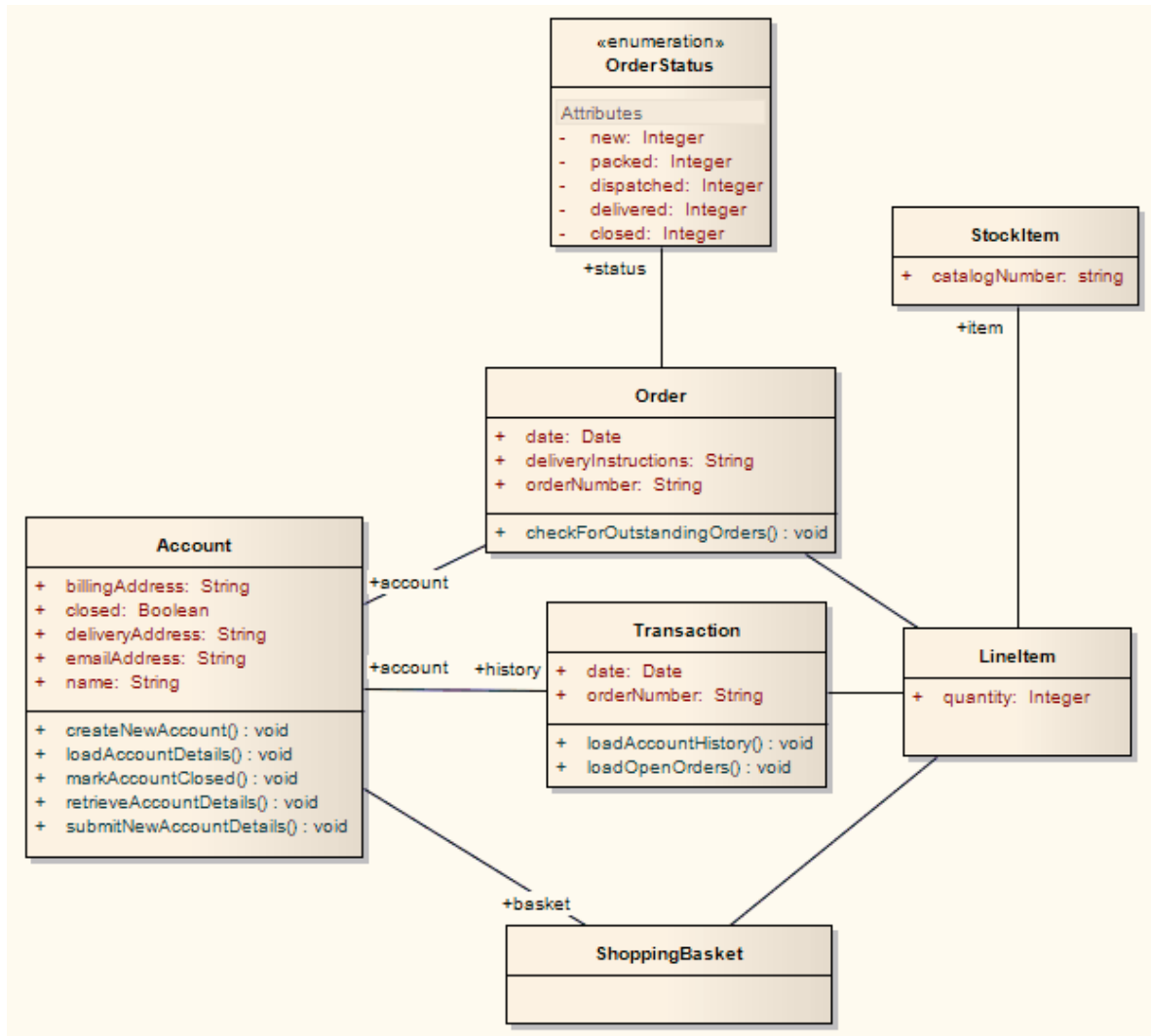
- [WSDL Model Structure](#) ²⁰⁷⁰
- [Model WSDL](#) ²⁰⁶⁹

9.2.10 XSD Transformation

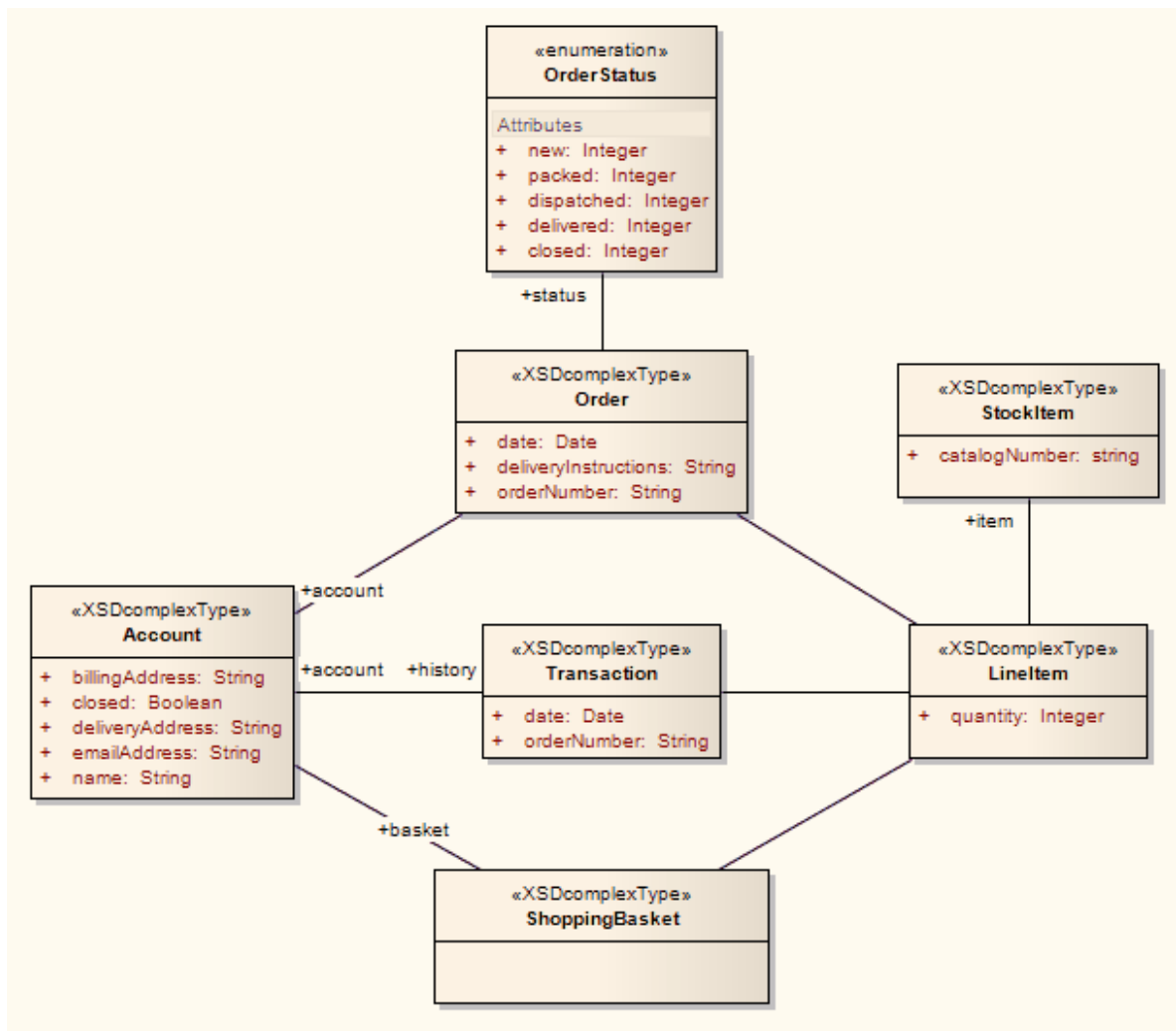
The purpose of the **XSD transformation** is to convert Platform-Independent Model (PIM) elements to UML Profile for XML elements as an intermediary step to creating an XML Schema. Each selected PIM Class element is converted to an «XSDcomplexType» element.

Example

The Platform-Independent Model (PIM):



After transformation becomes the PSM:



Which in turn generates the following XSD:

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:element name="Account" type="Account" />
  <xs:complexType name="Account" >
    <xs:sequence>
      <xs:element name="name" type="xs:string" />
      <xs:element name="billingAddress" type="xs:string" />
      <xs:element name="emailAddress" type="xs:string" />
      <xs:element name="closed" type="xs:boolean" />
      <xs:element name="deliveryAddress" type="xs:string" />
      <xs:element ref="Order" />
      <xs:element ref="ShoppingBasket" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="LineItem" type="LineItem" />
  <xs:complexType name="LineItem" >
    <xs:sequence>
      <xs:element name="quantity" type="xs:integer" />
      <xs:element ref="StockItem" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Order" type="Order" />
  <xs:complexType name="Order" >
    
```

```

        <xs:sequence>
            <xs:element name="date" type="xs:date"/>
            <xs:element name="deliveryInstructions" type="xs:string"/
>
            <xs:element name="orderNumber" type="xs:string"/>
            <xs:element ref="LineItem"/>
            <xs:element name="status" type="OrderStatus"/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="OrderStatus">
        <xs:restriction base="xs:string">
            <xs:enumeration value="new"/>
            <xs:enumeration value="packed"/>
            <xs:enumeration value="dispatched"/>
            <xs:enumeration value="delivered"/>
            <xs:enumeration value="closed"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:element name="ShoppingBasket" type="ShoppingBasket"/>
    <xs:complexType name="ShoppingBasket">
        <xs:sequence>
            <xs:element ref="LineItem"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="StockItem" type="StockItem"/>
    <xs:complexType name="StockItem">
        <xs:sequence>
            <xs:element name="catalogNumber" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="Transaction" type="Transaction"/>
    <xs:complexType name="Transaction">
        <xs:sequence>
            <xs:element name="date" type="xs:date"/>
            <xs:element name="orderNumber" type="xs:string"/>
            <xs:element ref="Account"/>
            <xs:element ref="LineItem"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>

```

Learn more

- For more information, see the [Generate XSD](#) ^[2049] topic

9.3 Transformation Templates

The Transformation Template mechanism is based very strongly on the Code Generation Template mechanism. For further information on Transformation Templates see the *Code Template Editor* section and also - for information on the Common Code Editor and intellisense - the *Editing Source Code* topic.

Access [Settings | Transformation Templates](#)

Reference

On the Transformation Templates Editor screen:

Field	Usage	See also
Language	Select the name of the transformation.	
Template	Display the contents of the active template; provide the editor for modifying templates.	Write Transformation S ^[1712] The Code Template Editor in MDG Development ^[1462] Code Template Syntax ^[1406]
Templates	List the base transformation templates. The active template is highlighted; the Modified field indicates whether you have changed the default template for the current transformation.	The Code Template Editor in MDG Development ^[1462]
New Transformation Type	Create a new transformation.	
Stereotype Overrides	List the stereotyped templates, for the active base template. The Modified field indicates whether you have modified a default stereotyped template.	
Add New Stereotyped Override	Invoke a dialog for adding a stereotyped template, for the currently selected base template.	
Add New Custom Template	Invoke a dialog for creating a custom stereotyped template.	
Help	Launch the Enterprise Architect Help topic for this dialog.	

Field	Usage	See also
Get Default Template	Update the editor display with the default version of the active template.	
Save	Overwrite the active templates with the contents of the editor.	
Delete	If you have overridden the active template, delete the override and replace it with the corresponding default transformation template.	

Learn more

- [Code Template Framework](#)^[1394] (Developing and Editing)
- [Code Template Editor](#)^[1404]
- [Code Template Framework](#)^[1924] (Using)
- [Editing Source Code](#)^[1799]
- [Export Code Generation and Transformation Templates](#)^[1400]
- [Import Code Generation and Transformation Templates](#)^[1401]

Learning Center topics

- **(Alt + F1) | Model Transformations | Transform | Write Transforms**

9.4 Write Transformations

This topic provides help in writing your own transformations. Transformations are based on the Code Generation Template Framework, therefore it is suggested that you read and understand the topics covering Code Generation Templates prior to using the Transformation Template language.

Access [Settings | Transformation Templates](#) (Ctrl+Alt+H)

Topics

Subjects concerning Transformation Templates include:

Topic	Detail	See also
Default Transformation Templates	Enterprise Architect provides a set of default transformation templates to customize to your requirements.	Default Transformation Templates ^[1713]
Intermediary Language Debugging	How to debug transformation scripts by checking the intermediary code generated from the Transform script.	Intermediary Language Debugging ^[1714]
Editing Source Code	Describes the facilities of the common Code Editor.	Editing Source Code ^[1799]
Code Template Framework	Provides forward engineering of UML models	Code Template Framework ^[1924] (Using) Code Template Framework ^[1394] (Modifying)
General Syntax for the Intermediary Language	Describes the grammar for the transformation.	Intermediary Language ^[1714]
Syntax for Creating Objects	Describes the syntax followed in creating objects or elements.	Syntax for Creating Objects ^[1718]
Syntax for Creating Connectors	Describes the syntax followed in creating connectors.	Syntax for Creating Connectors ^[1722] Transform Connectors ^[1724]
Transforming Duplicate Information	Describes how information is copied during transformation.	Transforming Duplicate Information ^[1727]

Topic	Detail	See also
Converting Types	Describes methods for converting data types to different target platform types.	Converting Types ^[1728]
Converting Names	Describes methods for converting names of elements to different target platform naming conventions.	Converting Names ^[1728]
Cross References	Provides cross verification of transformed elements.	Cross References ^[1730]

Notes

- Further hints and tips can be gleaned from a close study of the Transformation Templates provided with Enterprise Architect
- The Transformation Template editor applies the facilities of the common Code Editor
- Transformation Templates are accessed from the **Settings | Transformation Templates** menu option

Learn more

- [Code and Transform Templates](#) ^[1395]

Learning Center topics

- **(Alt + F1) | Model Transformations | Transform | Write Transforms**

9.4.1 Default Transformation Templates

In most transformations, there is a lot of information that is simply copied to the target model. In order to make writing new transformations simpler, Enterprise Architect provides a default set of transformation templates. These templates perform a simple copy of the source model to the target model. This means that in order to write a new transformation you can modify the default templates to make the required changes.

Notes

- When creating a new transformation you must modify at least one template before the transformation becomes available

Learn more

- [Write Transformations](#) ^[1712]
- [Transformation Templates](#) ^[1710]

9.4.2 Intermediary Language

All transformations in Enterprise Architect work by generating an intermediary text form of the model to generate.

Topics

Topic	Detail	See also
Accessing the Intermediary Language	You can access the intermediary language file using an external editor.	Intermediary Language Debugging ^[1714]
Usage	<p>Any object is represented in this language by the type of object (for example, Class, Action, Method, Generalization or Tag) followed by the properties of the object and the features that it is made from; the grammar for this resembles the following:</p> <pre> element : elementName { (elementProperty element) * } elementProperty : packageName stereotype propertyName = " propertyValueSymbol * " packageName : name = " propertyValueSymbol * " (. " propertyValueSymbol * ") * stereotype : stereotype = " propertyValueSymbol * " (, " propertyValueSymbol * ") * propertyValueSymbol : \ \ \ " Any character except " (U+0022), \ (U+005C) </pre> <ul style="list-style-type: none"> • <i>elementName</i> is any one of the set of element types • <i>propertyName</i> is any one of the set of properties <p>Literal strings can be included in property values by escaping a quote character</p> <pre>default = "\" Some string value.\""</pre>	Objects ^[1716] Connectors ^[1721]

9.4.3 Intermediary Language Debugging

Access **Project Browser package context menu | Transform Current Package (Ctrl+Shift+H)**

The script from an MDA template produces intermediate language text. However, on generating the model this script could return errors.

When an error occurs, the generated text can be viewed externally, preferably in an editor that prompts on updates to the file alterations.

Example

For a MySQL database, the template code might resemble the following:

```
$enumFieldName = "test"

Column
{
  name=%qt%%CONVERT_NAME($enumFieldName, "Pascal Case", "Camel Case")%qt%
  type=%qt%%CONVERT_TYPE(genOptDefaultDatabase, "Enum")%qt%
}
```

This returns the output in the generated text file as:

```
Column
{
  name = "test"
  type = "ENUM"
}
```

If there was an error in the original transform, such as a spelling error - 'Colum' - clicking the **Do Transform** button returns an error message referring to the first line of intermediate code that includes the error 'Colum'.

How to

When errors are returned on generating altered code, to debug the problem follow the steps below:

Step	Description	See also
1	Select the package to be transformed, and the Transform Current Package context menu. The Model Transformations dialog displays.	Transform Elements ^[1683]
2	In the Name column, select the checkbox against the type of transformation being altered.	
3	In the Intermediary File field, click on the (...) button and set the file location into which to generate the code.	
4	Select the Write Always checkbox, and click on the Write Now button to generate the script. This only generates the script, not the model.	
5	If an error is returned specifying the line number of the problem, open the file in an external Code Editor (with Line Numbering) and locate the line number of problem.	

Step	Description	See also
6	Alter the template code to correct the error.	
7	Click on the Do Transform button to check that the alteration corrected the problem.	

9.4.4 Objects

Topics

Topic	Detail	See also
Objects	<p>Objects are created in Enterprise Architect by generating text in the following form:</p> <pre> object Type { object Properties* XRef { xref } * Tag { tag } * Attribute { attributes } * Operation { operations } * Classifier { classifiers } * Parameter { parameters } * } </pre> <p>For example:</p> <pre> Class { name = "Example" language = "C++" Tag { name = "defaultCollectionClass" value = "List" } Attribute { name = "count" type = "int" } } </pre> <p>Every object created in a transformation should include an <i>XRef</i> element, as it enables Enterprise Architect to synchronize with the element and makes it possible to create a connector to that Class in a transformation.</p>	<p>objectTypes ^[1718]</p> <p>objectProperties ^[1718]</p> <p>XRef ^[1721]</p> <p>Tags ^[1720]</p> <p>Attributes ^[1719]</p> <p>Operations ^[1719]</p> <p>Parameters ^[1720]</p> <p>XRef ^[1718]</p>
objectTypes	<p>obj ect Type is one of the following object types:</p> <ul style="list-style-type: none"> • <i>Action</i> • <i>ActionPin</i> • <i>Activity</i> • <i>ActivityParameter</i> 	

Topic	Detail	See also
	<ul style="list-style-type: none"> • <i>ActivityPartition</i> • <i>ActivityRegion</i> • <i>Actor</i> • <i>Association</i> • <i>Change</i> • <i>Class</i> • <i>Collaboration</i> • <i>CollaborationUse</i> • <i>Component</i> • <i>DeploymentSpecification</i> • <i>DiagramFrame</i> • <i>Decision</i> • <i>EntryPoint</i> • <i>Event</i> • <i>ExceptionHandler</i> • <i>ExecutionEnvironment</i> • <i>ExitPoint</i> • <i>ExpansionNode</i> • <i>ExpansionRegion</i> • <i>ExposedInterface</i> • <i>GUIElement</i> • <i>InteractionFragment</i> • <i>InteractionOccurrence</i> • <i>InteractionState</i> • <i>Interface</i> • <i>InterruptibleActivityRegion</i> • <i>Issue</i> • <i>Iteration</i> • <i>Object</i> • <i>ObjectNode</i> • <i>MessageEndpoint</i> • <i>Node</i> • <i>Package</i> • <i>Parameter</i> • <i>Part</i> • <i>Port</i> 	

Topic	Detail	See also
	<ul style="list-style-type: none"> • <i>ProvidedInterface</i> • <i>RequiredInterface</i> • <i>Requirement</i> • <i>Sequence</i> • <i>State</i> • <i>StateMachine</i> • <i>StateNode</i> • <i>Synchronization</i> • <i>Table</i> • <i>TimeLine</i> • <i>Trigger</i> • <i>UMLDiagram</i> • <i>UseCase</i> 	
objectProperties	<p>obj ect Pr oper t i es is zero, or one instance of one or more of the following properties:</p> <ul style="list-style-type: none"> • <i>Abstract</i> • <i>Alias</i> • <i>Arguments</i> • <i>Author</i> • <i>Cardinality</i> • <i>Classifier</i> • <i>Complexity</i> • <i>Concurrency</i> • <i>Filename</i> • <i>Header</i> • <i>Import</i> • <i>IsActive</i> • <i>IsLeaf</i> • <i>IsRoot</i> • <i>IsSpecification</i> • <i>Keyword</i> • <i>Language</i> • <i>Multiplicity</i> • <i>Name</i> • <i>Notes</i> 	

Topic	Detail	See also
	<ul style="list-style-type: none"> • <i>Persistence</i> • <i>Phase</i> • <i>Scope</i> • <i>Status</i> • <i>Stereotype</i> • <i>Version</i> • <i>Visibility</i> 	
Attributes	<p>Attributes are created with the same structure as objects, and include the following properties:</p> <ul style="list-style-type: none"> • <i>Alias</i> • <i>Classifier</i> • <i>Collection</i> • <i>Container</i> • <i>Containment</i> • <i>Constant</i> • <i>Default</i> • <i>Derived</i> • <i>LowerBound</i> • <i>Name</i> • <i>Notes</i> • <i>Ordered</i> • <i>Scope</i> • <i>Static</i> • <i>Stereotype</i> • <i>Type</i> • <i>UpperBound</i> • <i>Volatile</i> <p>and the following elements:</p> <ul style="list-style-type: none"> • <i>Classifier</i> • <i>Tag</i> • <i>XRef</i> 	
Operations	<p>Operations are created with the same structure as objects, and include the following properties:</p>	

Topic	Detail	See also
	<ul style="list-style-type: none"> • <i>Abstract</i> • <i>Alias</i> • <i>Behavior</i> • <i>Classifier</i> • <i>Code</i> • <i>Constant</i> • <i>IsQuery</i> • <i>Name</i> • <i>Notes</i> • <i>Pure</i> • <i>ReturnArray</i> • <i>Scope</i> • <i>Static</i> • <i>Stereotype</i> • <i>Type</i> <p>and the following elements:</p> <ul style="list-style-type: none"> • <i>Classifier</i> • <i>Parameter</i> • <i>Tag</i> • <i>XRef</i> 	
Parameters	<p>Parameters are created with the same structure as objects, and include the Tag element and the following properties:</p> <ul style="list-style-type: none"> • <i>Classifier</i> • <i>Default</i> • <i>Fixed</i> • <i>Name</i> • <i>Notes</i> • <i>Kind</i> • <i>Stereotype</i> 	
Tags	<ul style="list-style-type: none"> • <i>Name</i> • <i>Value</i> 	

Special Cases:

The following table represents some special cases of object definitions.

Topic	Detail	See also
Packages	<p>Packages differ from other objects in the following ways:</p> <ul style="list-style-type: none"> • A reduced set of properties of <i>alias</i>, <i>author</i>, <i>name</i>, <i>namespaceRoot</i>, <i>notes</i>, <i>scope</i>, <i>stereotype</i> and <i>version</i> • The extra property <i>namespaceRoot</i> • Must have a name specified • <i>Name</i> can be a qualified name; when a qualified name is specified the properties given are applied only to the final package • Can contain other packages • Can't contain attributes and operations 	
XRef	<p>Cross references are defined using the transform statements. The properties include:</p> <ul style="list-style-type: none"> • Namespace • Name • Source • Notes 	Cross References <small>[1730]</small> XRef element <small>[1730]</small>
Tables	<p>Tables are a special sort of object, with the following differences from other object types:</p> <ul style="list-style-type: none"> • Can include columns and primary keys • Cannot include attributes 	Foreign Key <small>[1726]</small>
Columns	<p>Columns are similar to attributes, but have an <i>autonumber</i> element containing <i>Startnum</i> and increment, and the following added properties:</p> <ul style="list-style-type: none"> • <i>Length</i> • <i>NotNull</i> • <i>Precision</i> • <i>PrimaryKey</i> • <i>Scale</i> • <i>Unique</i> <p>In the column definition, you cannot assign a value to the <i>NotNull</i>, <i>PrimaryKey</i> or <i>Unique</i> properties.</p>	

Topic	Detail	See also

9.4.5 Connectors

Creating connectors in a transformation can be complex, but the process has the same form as creating elements. The difference is that you must also specify each end.

Topics

Topic	Detail	See also
Intermediary Language	<p>Connectors are represented in the Intermediary language as:</p> <pre> Connector Type { connector Properties* Source {sourceProperties*} Target {targetProperties*} } </pre> <p>For example:</p> <pre> Association { name="anAssociation" stereotype="" direction="Unspecified" Source { access="Private" navigability="Unspecified" } Target { access="Private" multiplicity="1..*" } } </pre>	
Connector Types	<p>The different connectors that can be created are:</p> <ul style="list-style-type: none"> • Aggregation • Assembly • Association • Collaboration • ControlFlow • Connector • Delegate • Dependency • Deployment • ForeignKey 	

Topic	Detail	See also
	<ul style="list-style-type: none"> • Generalization • InformationFlow • Instantiation • Interface • InterruptFlow • Manifest • Nesting • NoteLink • ObjectFlow • Package • Realization • Sequence • Transition • UseCase • Uses 	
Connector Properties	<p><code>connect</code> or <code>Properties</code> is zero, or one instance of one or more of the following properties:</p> <ul style="list-style-type: none"> • alias • direction • notes • name • stereotype • XRef 	
Source and Target Properties	<p><code>sourceProperties</code> and <code>targetProperties</code> are zero, or one instance of one or more of the following properties:</p> <ul style="list-style-type: none"> • aggregation • alias • allowduplicates • changeable • constraint • containment • navigability • membertype 	

Topic	Detail	See also
	<ul style="list-style-type: none"> • multiplicity • Notes • ordered • qualifier • role • scope • stereotype • XRef 	

Notes

- Each connector is transformed at both end objects, therefore the connector might appear twice in the transformation; this is nothing to be concerned about, but you should check carefully that the connector is generated exactly the same way, regardless of which end is on the current Class

Learn more

- [Transform Connectors](#) [1724]
- [Transform Foreign Keys](#) [1726]

9.4.6 Transform Connectors

There are two different types of Class that you can use as a connector end: one created by a transformation, and one for which you already know the GUID.

Topics

Topic	Detail	See also
<p>Connect to a Class Created by a Transformation</p>	<p>The most common connection is to connect to a Class created by a transformation; to do this you must have three items of information:</p> <ul style="list-style-type: none"> • The original Class GUID • The name of the transformation • The name of the transformed Class <p>This type of connector is created using the TRANSFORM_REFERENCE function macro; when the element is in the current transformation, it can be safely omitted from the transformation.</p> <p>The simplest example of this is when you have created multiple Classes from a single Class in a transformation and want a connector between them; consider this example from the EJB Entity transformation:</p> <pre> Dependency { </pre>	<p>Cross References <small>[1730]</small></p>

Topic	Detail	See also
	<pre> %TRANSFORM_REFERENCE(" EJBReal i zeHome" , classGUI D) % st ereot ype=" EJBReal i zeHome" Source { %TRANSFORM_REFERENCE(" EJBEnt i tyBean" , classGUI D) % } Tar get { %TRANSFORM_REFERENCE(" EJBHomeI nt er face" , classGUI D) % } } </pre> <p>There are three uses of the TRANSFORM_REFERENCE macro: one to identify this connector for synchronization purposes and the other two to identify the ends; all three use the same source GUID, because they all come from the one original Class.</p> <p>None of the three have to specify the transformation because the two references are referencing something in the current transformation - each of them then only has to identify the transform name.</p> <p>Of course, it is also possible to create a connector from another connector.</p> <p>You can create a connector template and list over all connectors connected to a Class from the Class level templates; you don't have to worry about only generating it once, because if you have created a TRANSFORM_REFERENCE for the connector then Enterprise Architect automatically synchronizes them.</p> <p>The following copies the source connector:</p> <pre> %connector Type% { %TRANSFORM_CURRENT() % %TRANSFORM_REFERENCE(" Connector" , connector GUI D) % Source { %TRANSFORM_REFERENCE(" Cl ass" , connector SourceGUI D) % %TRANSFORM_CURRENT(" Source") % } Tar get { %TRANSFORM_REFERENCE(" Cl ass" , connector Dest GUI D) % %TRANSFORM_CURRENT(" Tar get ") % } } </pre>	<p>Copy Information</p> <p>1727</p>
<p>Connecting to a Class for which you know the GUID</p>	<p>The second type of Class that you can use as a connector end is one for which you know the current GUID.</p> <p>To do this, specify the GUID of the target Class in either the source or target end; the following example creates a dependency from a Class created in a transformation, to the Class it was transformed from:</p> <pre> Dependency { </pre>	

Topic	Detail	See also
	<pre> %TRANSFORM_REFERENCE(" SourceDependency" , classGUI D) % stereotype=" transformedFrom" Source { %TRANSFORM_REFERENCE(" Cl ass" , cl assGUI D) % } Target { GUI D=%qt %%cl assGUI D%%qt % } </pre>	

Notes

- Each connector is transformed at both end objects, therefore the connector might appear twice in the transformation; this is nothing to be concerned about, but you should check carefully that the connector is generated exactly the same way, regardless of which end is on the current Class

Learn more

- [Connectors](#)^[1722]

9.4.7 Transform Foreign Keys

ForeignKey is a special case where not just a connector is created - you must also list the columns involved in the transformation.

In addition, tags specified on the connector are actually created on the foreign key operation in the source Class, and a cascade property can be added `cascade=" update" , " delete" .`

The following example is from the DDL transform “Connector” template.

For the Template:

```

For ei gnKey
{
  %TRANSFORM_REFERENCE("FK1", connectorGUI D)%
  Source
  {
    %TRANSFORM_REFERENCE("Li nkTabl e", connectorGUI D)%
    name=%qt%%connectorDestEl emName%%qt%
    Col umn
    {
      name=%qt%%CONVERT_NAME(connectorDestEl emName, "Pascal Case", "Camel Case"
)%ID%qt%
      type=%qt%%CONVERT_TYPE(genOptDefaul tDatabase, "Integer")%%qt%
    }
  }
  Target
  {
    %TRANSFORM_REFERENCE("Tabl e", connectorDestEl emGUI D)%
    Col umn
    {
      name=%qt%%CONVERT_NAME(connectorDestEl emName, "Pascal Case", "Camel Case"
)%ID%qt%
      type=%qt%%CONVERT_TYPE(genOptDefaul tDatabase, "Integer")%%qt%
    }
  }
}

```

```

    }
  }
}

```

The following code is generated for a Foreign Key:

```

ForeignKey
{
  XRef { namespace=" DDL " name=" FK1 " source="{ 18ED24EA- 0669- 499c- A066-
FABEE6772ED1} " }
  Source
  {
    XRef { namespace=" DDL " name=" Table " source="{ CC6A084C- 9B56- 4a1e-
8196- C66A74E59B66} " }
    name=" FK_account "
    multiplicity=" 0.. * "
    Column
    {
      name=" account "
      type=" I N T E G E R "
    }
  }
  Target
  {
    XRef { namespace=" DDL " name=" Table " source="{ 791AEAB5- 7831- 4d2a-
92C7- 198C10B31AF0} " }
    multiplicity=" 1 "
    Column
    {
      name=" account I D "
      type=" I N T E G E R "
    }
  }
}
}

```

9.4.8 Copy Information

In many transformations there is a substantial amount of information to be copied.

It would be tedious to type all of the common information into a template so that it is copied to the transformed Class; the alternative is to use the **TRANSFORM_CURRENT** and **TRANSFORM_TAGS** function macros.

Reference

Topic	Detail	See also
Copy Object	TRANSFORM_CURRENT(<listOfExcludedItems>) The function generates an exact copy of all the properties of the current item, except for the items named in <listOfExcludedItems>.	Objects [1718]
Copy Connector	Another form of the function is available when transforming connectors that enables either end of the connector to be copied: TRANSFORM_CURRENT(<connectorEnd>,<listOfExcludedItems>) This generates an exact copy of the connector end specified by <connectorEnd> (either Source or Target) except for the items named in <listOfExcludedItems>.	Connectors [1722]

Topic	Detail	See also
Copy Tags	<p>TRANSFORM_TAGS(<listOfExcludedItems>)</p> <p>The function generates an exact copy of all the Tagged Values of the current item, except for the items named in <listOfExcludedItems>.</p>	

9.4.9 Convert Types

Topics

Topic	Detail	See also
Synopsis	<p>Different target platforms almost certainly require different data types, so you often require a method of converting between types, as offered by the following macro:</p> <ul style="list-style-type: none"> • CONVERT_TYPE(<destinationLanguage>, <originalType>) <p>This function converts <originalType> to the corresponding type in <destinationLanguage>, using the datatypes and common types defined in the model, where <originalType> is assumed to be a platform independent common type.</p> <p>A similar macro is available when transforming common datatypes to the datatypes for a specified database:</p> <ul style="list-style-type: none"> • CONVERT_DB_TYPE(<destinationDatabase>, <originalType>) <p>This function converts <originalType> to the corresponding datatypes in <destinationDatabase>, which is defined in the model; <originalType> refers to a platform independent common datatype.</p>	

9.4.10 Convert Names

Different target platforms use different naming conventions. As a result you might not want to copy the names of your elements directly into the transformed models. To facilitate this requirement, Enterprise Architect's transformation templates provide a CONVERT_NAME function macro.

Another way in which you can transform a name is to remove a prefix from the original name, with the REMOVE_PREFIX macro.

Topics

Topic	Detail	See also
CONVERT_NAME <originalName>	This macro converts <originalName>, which is assumed to be in	

Topic	Detail	See also
<p><originalFormat>, <targetFormat></p>	<p><i><originalFormat></i>, to <i><targetFormat></i>.</p> <p>The supported formats are:</p> <ul style="list-style-type: none"> • Camel Case: New words start with a capital letter except for the first word, which begins with a lower case letter; for example, <i>myVariableTable</i> • Pascal Case: Same as Camel Case but the first letter of the first word is upper case; for example, <i>MyVariableTable</i> • Spaced: Words are separated by spaces; the case of letters is ignored • Underscored: Words are separated by underscores; the case of letters is ignored. <p>The original format might also specify a list of delimiters to be used. For example a value of '_' breaks words whenever either a space or underscore is found.</p> <p>The target format might also use a format string that specifies the case for each word and a delimiter between them. It takes the following form:</p> <p><i><firstWord></i>(<i><delimiter></i>)<i><otherWords></i></p> <ul style="list-style-type: none"> • <i><firstWord></i> controls the case of the first word (see below) • <i><delimiter></i> is the string generated between words • <i><otherWords></i> applies to all words after the first word <p><i><firstWord></i> and <i><otherWords></i> are both a sequence of two characters. The first character represents the case of the first letter of that word, and the second character represents the case of all subsequent letters. An upper case letter forces the output to upper case, a lower case letter forces the output to lower case, and any other character preserves the original case.</p> <p>Example 1: To capitalize the first letter of each word and separate multiple words with a space:</p> <p style="padding-left: 40px;">" Ht () Ht " to output " My Var i abl e Tabl e "</p> <p>Example 2: To generate the equivalent of Camel Case, but reverse the roles of upper and lower case; that is, all characters are upper case except for the first character of each word <i>after</i> the first word:</p> <p style="padding-left: 40px;">" HT() hT " to output " MY vARI ABLE t ABLE "</p>	

Topic	Detail	See also
REMOVE_PREFIX (<originalName> , <prefixes>)	<p>This macro removes any prefix found in <i><prefixes></i> from <i><originalName></i>. The prefixes are specified in a semi-colon separated list.</p> <p>The macro is often used in conjunction with the <i>CONVERT_NAME</i> macro. For example, this code creates a <i>get property name</i> according to the options for Java.</p> <pre>\$propertyName=%REMOVE_PREFIX(attributeName, genOpt PropertyPrefix) % %if genOpt GeneralisedProperties==" T" % \$propertyName=%CONVERT_NAME (\$propertyName, " camel case", " pascal case") % %endif %</pre>	

Notes

- Acronyms are not supported when converting from Camel Case or Pascal Case

9.4.11 Cross References**Topics**

Topic	Detail	See also
Abstract	<p>Cross References are an important part of transformations. They are used to:</p> <ul style="list-style-type: none"> • Find the transformed Class to synchronize with • Create connectors between transformed Classes • Specify a classifier of a type • Determine where to transform to for future transformations <p>Each cross reference has three different parts:</p> <ul style="list-style-type: none"> • A <i>Namespace</i>, corresponding to the transformation that generated the element • A <i>Name</i>, which is a unique reference to something that can be generated in the above transformation • A <i>Source</i>, which is the GUID of the element that this element was created from <p>When writing the templates for a transformation, it is easiest to create the cross references using the TRANSFORM_REFERENCE macro that is defined for this purpose. It has three optional parameters.</p> <p>TRANSFORM_REFERENCE(<name>, <sourceGuid>, <namespace>)</p> <p>This generates a reference that can be used in the ways described above.</p>	<p>DDL Transformation [1689]</p> <p>XRef element [1730]</p>

Topic	Detail	See also
	<p>It resembles the following.</p> <pre>XRef { namespace=" <namespace>" name=" <name>" source=" <sourceGuid>"</pre> <p>Where:</p> <ul style="list-style-type: none"> • If <i><name></i> is not specified it gets the name of the current template • If <i><sourceGUID></i> is not specified it gets the GUID of the current Class • If <i><namespace></i> is not specified it gets the name of the current transformation <p>The only time that this should be specified is when creating a connector to a Class created in a different transformation.</p> <p>A good example of the use of cross references is in the DDL templates provided with Enterprise Architect. In the Class template a cross reference is created with the name "Table". Then up to two different connectors can be created, each of which must identify the two Classes it connects using cross references while having its own unique cross reference.</p>	
Specify Classifiers	<p>Objects, attributes, operations and parameters can all reference another element in the model as their type. When this type is created from a transformation you must use a cross reference to specify it, using the TRANSFORM_CLASSIFIER macro.</p> <p>TRANSFORM_CLASSIFIER(<name>, <sourceGuid>, <namespace>)</p> <p>Generates a cross reference within a classifier element, where the parameters are identical to the TRANSFORM_REFERENCE macro but the name Classifier is generated instead of <i>XRef</i>.</p> <p>If the target classifier already exists in the model before the transformation, a TRANSFORM_CLASSIFIER is inappropriate and instead the GUID can be given directly to a classifier attribute.</p> <p>If a classifier is specified for any type it overrides the type specified.</p>	

Part



10 Software Engineering



Software Engineering is the process of designing, implementing and maintaining software. This section describes how Enterprise Architect can assist in this process by providing a design environment and tools including **automated code generation**, **reverse engineering** of source code and **synchronization** between the source code and model.

Topics

Topic	Link
Software Design	GoF Patterns ^[1928]
Development Environment	Development Tools ^[1734] Code Build & Debug ^[1824] MDG Integration and Code Engineering ^[1740]
Generate Source Code	Generate Source Code ^[1768]
Importing Source Code	Importing Source Code ^[1790]

Learn more

- Enterprise Architect provides a set of tools that goes beyond traditional debugging capabilities, see [Visual Execution Analyzer](#) ^[2143]
- [Database Engineering](#) ^[1931] is also available, including keys, triggers, constraints, RI and other relational database features, for and from a range of database products
- Enterprise Architect provides technologies to support Service Oriented Architecture; see [SOA and XML Engineering](#) ^[2026]

10.1 Development Tools



Topics

Topic	Detail	See also
A Tightly Integrated Development Environment with Outstanding Tools and Functionality	<ul style="list-style-type: none"> • Design and elaborate your Class model in UML and other technologies • Generate code and reverse engineer existing code • Customize code generation with templates • Invoke compilers and other command line tools to build, run and deploy your application • Use the built in source code editor to write and maintain source files • Use the built in debugger to inspect and correct code at run time • Use the Profiler to fine tune Windows native C/C++ code • Use the visualization tools to generate Sequence diagrams from executing code • Use the testing capabilities to validate execution with Class- and method-level constraints and invariants • Link to junit and NUnit test cases • Link or integrate with Eclipse or Visual Studio when necessary to further enhance your development process 	
Abstract	<p>Enterprise Architect (Professional edition and above) provides an extraordinary range of features and a unique, rich, tightly coupled toolset for rapidly developing, visualizing, debugging, testing, maintaining and generally working with even the most complex source code.</p> <p>A wide range of standard programming languages are supported in design and reverse engineering; inbuilt editors with custom syntax highlighting and tight integration with the model provide a powerful and effective programming environment; coupled with the build/debug and testing capabilities, Enterprise Architect is both a remarkable analysis and design tool, and a formidable debug and testing environment.</p> <p>Enterprise Architect's Visual Execution Analyzer also supports Java, .NET and native Windows applications written in C/C++ using Microsoft's compilers; in addition to powerful tools for standard debugging, Enterprise Architect supports recording Sequence diagrams from executing code, capturing of method calls and conversion into custom call sets for testing purposes, validation of execution against State diagrams, and more.</p> <p>The Model Driven Development Environment (MDDE) provides one of the</p>	

Topic	Detail	See also
	<p>richest and most useful feature sets available for working with source code and ensuring the construction of robust and effective applications; the MDDE integrates code and model by providing options to either generate source code from the model or reverse engineer existing source code into a model - source code and model can be synchronized in either direction.</p>	
Application Patterns	<p>Enterprise Architect provides complete starter projects, including model information, code and build scripts, for several basic application types including:</p> <ul style="list-style-type: none"> • MFC Windows applications • Java programs • ASP.NET web services 	Generate Application Pattern <small>[1738]</small>
Relationships of Software Engineering Units	<pre> graph TD Design[Design] --> CG[Code Generation] CG --> SC1[/Source Code/] SC1 --> Import[Import] SC1 --> CE[Code Editing] Import --> CE CE --> Build[Build] Build --> Test[Test] Build --> Debug[Debug] Build --> Run[Run] Build --> Deploy[Deploy] </pre>	
Supported Languages	<p>Popular languages supported include:</p> <ul style="list-style-type: none"> • C/C++ • Java • Microsoft .NET family • ADA • Python • Perl <p>Toolboxes provide for different modeling technologies.</p>	

Notes

- Although you can generate and reverse engineer code in a range of languages, Execution Analysis debugging and recording are supported for the following platforms / languages only:

- Microsoft Windows Native C
- Microsoft Windows Native C++
- Microsoft Windows Visual Basic
- Microsoft .NET Family (C#, J#, VB)
- Sun Microsystems Java

Learn more

- [Software Engineering](#)^[1733]
- [Getting Started](#)^[1825]
- [Analyzer Scripts](#)^[1827]
- [Editing Source Code](#)^[1799]
- [Debugging Actions](#)^[1874]

Learning Center topics

- (Alt + F1) | **Software Engineering**

10.1.1 Overview of Development

Code Engineering with Enterprise Architect broadly encompasses various processes for generating or transforming code from your UML model and importing code into the model, to support model development in several coding languages, database development and SOA development.

Topics

Topic	Detail	See also
Model Driven Code Engineering	<ul style="list-style-type: none"> • Source code generation and reverse engineering for many popular languages, including C++, C#, Java, Delphi, VB.Net, Visual Basic, ActionScript, Python and PHP • A built in 'syntax highlighting' source code editor • Code generation templates, which enable you to customize the generated source code to your company specifications 	Software Engineering ^[1733]
Transformations for Rapid Development	<ul style="list-style-type: none"> • Advanced Model Driven Architecture (MDA) transformations using transformation templates • Built-in transformations for DDL, C#, Java, EJB and XSD • One Platform Independent Model can be used to generate and synchronize multiple Platform Specific 	Model Transformations - MDA ^[1679]

Topic	Detail	See also
	Models, providing a significant productivity boost	
Visual Execution Analysis / Debugging, Verification and Visualization	<ul style="list-style-type: none"> • Execute build, test, debug, run and deploy scripts • Integrate UML development and modeling with source development and compilation • Generate NUnit and JUnit test Classes from source Classes using MDA Transformations • Integrate the test process directly into the Enterprise Architect IDE • Debug .NET, Java and Microsoft Native (C, C++ and Visual Basic) applications • Design and execute Test suites based on <i>Programming by Contract</i> principles 	Visual Execution Analysis ^[2143]
Database Modeling	<p>Enterprise Architect enables you to:</p> <ul style="list-style-type: none"> • Reverse engineer from many popular DBMSs, including SQL Server, My SQL, Access, PostgreSQL and Oracle 9i, 10g or 11g • Model database tables, columns, keys, foreign keys and complex relationships using UML and an inbuilt data modeling profile • Forward generate DDL scripts to create target database structures 	Database Engineering ^[1931]
XML Technology Engineering	<p>Enterprise Architect enables you to rapidly model, forward engineer and reverse engineer two key W3C XML technologies:</p> <ul style="list-style-type: none"> • XML Schema (XSD) • Web Service Definition Language (WSDL) <p>XSD and WSDL support is critical for the development of a complete Service Oriented Architecture (SOA), and the coupling of UML 2.3 and XML provides the natural mechanism for implementing XML-based SOA artifacts within an organization.</p>	SOA and XML ^[2026] XML Schema - XSD ^[2027] Web Services - WSDL ^[2067]

10.1.2 Application Patterns (Model + Code)

To get you going with a code based project as fast as possible, Enterprise Architect enables you to generate starter projects including model information, code and build scripts for one of several basic application types. Patterns include:

- MFC Windows applications
- Java programs
- ASP.NET web services

Access **Project | New Model (Ctrl+Shift+M) > Application Patterns**
Project Browser | Package context menu | Add | Add a New Model using Wizard > Application Patterns

Reference

Field	Usage	See also
Technology	Select the appropriate technology.	
Name	Displays the application patterns available for the selected technology; select the required pattern to import.	
description field	Displays a description of the selected pattern.	
Destination folder	Browse for and select the directory in which to load the source code for the application.	
Use Local Path	Enable the selection of an existing local path to place the source code under; changes the Destination folder field to a drop-down selection.	
Compiler command	Displays the default compiler command path for the selected technology; you must either: <ul style="list-style-type: none"> • Confirm that the compiler can be found at this path, or • Edit the path to the compiler location 	Local Paths <small>[1900]</small> Local Paths Dialog <small>[1901]</small>
Edit Local Paths	Many application patterns specify their compiler using a local path. The first time you use any pattern you must click on this button to ensure the local path points to the correct location. The Local Paths dialog displays.	

Notes

- If required, you can define custom application patterns by adding files to the *AppPatterns* directory where Enterprise Architect is installed; top level directories are listed as Technologies and can contain an icon file to customize the icon displayed for the technology

Directories below this are defined as groups in the patterns list; the patterns are defined by the presence of four files with a matching name: a zip file (.zip), XML file (.xml), config file (.cfg) and optional icon (.ico)

- The *config* file supports the following fields:
 - [provider], [language], [platform], [url], [description], [version] - all displayed in the description field
 - [xmireootpaths] - the root path of the source code in the exported xmi; this is replaced with the selected destination folder when the user applies the application pattern

Learn more

- [Model Wizard](#)^[660]

10.2 MDG Integration and Code Engineering

Topics

Topic	Detail	See also
Abstract/Usage	<p>MDG Integration for Eclipse and MDG Integration for Visual Studio are standalone products that provide an enhanced code engineering functionality between Enterprise Architect and the development environments.</p> <p>The MDG Integration programs provide a lightweight bridge between Enterprise Architect and the development environment, offering enhanced code generation, reverse engineering and synchronization between code and the UML model; merging changes can be achieved with minimal effort, and navigation between model and source code is significantly enhanced.</p> <p>Trial versions of MDG Integration for Eclipse and MDG Integration for Visual Studio can be downloaded from the Sparx Systems website.</p>	<p>www.sparxsystems.com/products/mdg/int/eclipse/index.html (Online Resource)</p> <p>www.sparxsystems.com/products/mdg/int/vs/index.html (Online Resource)</p>

10.3 Modeling Conventions



In order to get the most out of the round trip engineering in Enterprise Architect, you must be familiar with the modeling conventions used when generating and reverse engineering the languages you use. This topic describes the stereotypes, Tagged Values and other conventions used in code engineering in Enterprise Architect for the following supported languages:

Topic	Link
Action Script	ActionScript ^[1742]
Ada 2005 (Systems Engineering and Ultimate editions)	Ada 2005 ^[1743]
C	C ^[1745]
C#	C# ^[1748]
C++	C++ ^[1750]
Delphi	Delphi ^[1754]
Java	Java ^[1756]
PHP	PHP ^[1758]
Python	Python ^[1759]
SystemC (Systems Engineering and Ultimate editions)	SystemC ^[1760]
Verilog (Systems Engineering and Ultimate editions)	Verilog ^[1763]
VHDL (Systems Engineering and Ultimate editions)	VHDL ^[1764]
Visual Basic	Visual Basic ^[1766]
Visual Basic .NET	Visual Basic .NET ^[1767]

Notes

- Enterprise Architect incorporates a number of visibility indicators or scope values for its supported languages; these include, for:
 - All languages - Public (+), Protected (#) and Private (-)
 - Java - Package (~)
 - Delphi - Published (^)
 - C# - Internal (~), Protected Internal (^)
 - ActionScript - Internal (~)
 - VB.NET - Friend (~), Protected Friend (^)

- PHP - Package (~)
- Python - Package (~)
- C - Package (~)
- C++ - Package (~)

10.3.1 ActionScript Conventions

Enterprise Architect supports round trip engineering of ActionScript 2 and 3, where the following conventions are used.

Reference

Stereotypes

Stereotype	Applies To	Corresponds To
literal	Operation	A literal method referred to by a variable.
property get	Operation	A read property.
property set	Operation	A write property.

Tagged Values

Tag	Applies To	Corresponds To
attribute_name	Operation with stereotype <i>property get</i> or <i>property set</i>	The name of the variable behind this property.
dynamic	Class or Interface	The <i>dynamic</i> keyword.
final	ActionScript 3: Operation	The <i>final</i> keyword.
intrinsic	ActionScript 2: Class	The <i>intrinsic</i> keyword.
namespace	ActionScript 3: Class, Interface, Attribute, Operation	The namespace of the current element.
override	ActionScript 3: Operation	The <i>override</i> keyword.
prototype	ActionScript 3: Attribute	The <i>prototype</i> keyword.
rest	ActionScript 3: Parameter	The <i>rest</i> parameter (...).

Topics

Topic	Detail	See also
Common Conventions	<ul style="list-style-type: none"> • Package qualifiers (ActionScript 2) and Packages (ActionScript 3) are generated when the current package is not a namespace root • An unspecified type is modeled as <i>var</i> or an empty Type field 	Namespace s_[1777]

Topic	Detail	See also
ActionScript 3 Conventions	<ul style="list-style-type: none"> The <i>Is Leaf</i> property of a Class corresponds to the sealed keyword If a <i>namespace</i> tag is specified it overrides the <i>Scope</i> that is specified 	

Learn more

- [Import Source Code](#)^[1792]
- [Generate Source Code](#)^[1768]
- [ActionScript Options](#)^[1909]

10.3.2 Ada 2005

Enterprise Architect supports round trip engineering of Ada 2005, where the following conventions are used.

Reference

Stereotypes

Stereotype	Applies To	Corresponds To
adaPackage	Class	A package specification in Ada 2005 without a tagged record.
adaProcedure	Class	A procedure specification in Ada 2005.
delegate	Operation	Access to a subprogram.
enumeration	Inner Class	An <i>enum</i> type.
struct	Inner Class	A record definition.
typedef	Inner Class	A type definition, subtype definition, access type definition, renaming.

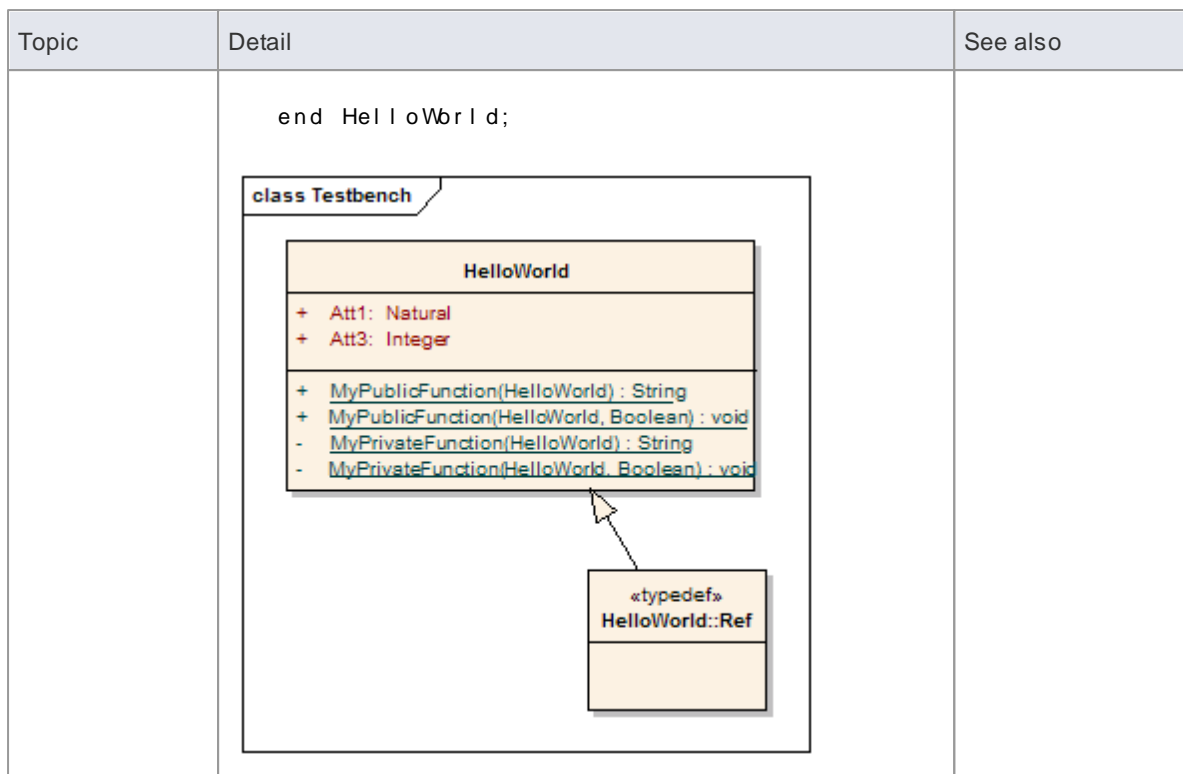
Tagged Values

Tag	Applies To	Corresponds To
Discriminant	Inner Class with stereotype <i>typedef</i>	The type's discriminant.
IsAccess	Parameter	Determination of whether the parameter is an access variable.
InstantiatedUnitType	Inner Class with stereotype <i>typedef</i>	The instantiated unit's type (<i>Package / Procedure / Function</i>).

Tag	Applies To	Corresponds To
PartType	Inner Class with stereotype <i>typedef</i>	The part type (<i>renames</i> or <i>new</i>).
Type	Inner Class with stereotype <i>typedef</i>	If Value = <i>Sub Type</i> , set subtype If Value = <i>Access</i> , set access type.

Topics

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> • Appropriate type of source files: Ada specification file, .ads • Ada 2005 imports packages defined as either <<adaPackage>>Class or Class, based on the settings in the Ada 2005 options • A package in the Ada specification file is imported as a Class if it contains a Tagged Record, the name of which is governed by the options Use Class Name for Tagged Record and Alternate Tagged Record Name; all attributes defined in that Tagged Record are absorbed as the Class's attributes • A procedure / function in an Ada specification file is considered as the Class's member function if its first parameter satisfies the conditions specified in the options Ref Param Style, Ignore Reference parameter name and Ref parameter name • The option Define Reference for Tagged Record, if enabled, creates a reference type for the Class, the name of which is determined by the option Reference Type Name <p>For example: HelloWorld.ads</p> <pre> package HelloWorld is type HelloWorld is tagged record Att1: Natural; Att3: Integer; end record; -- Public Functions function MyPublicFunction (P: HelloWorld) return String; procedure MyPublicFunction (P1: in out HelloWorld; AFlag: Boolean); private -- Private Functions function MyPrivateFunction (P: HelloWorld) return String; procedure MyPrivateFunction (P1: in out HelloWorld; AFlag: Boolean); </pre>	Ada 2005 Options [1908]



Notes

Ada 2005 support is available in the System Engineering and Ultimate editions of Enterprise Architect.

Learn more

- [Import Source Code](#)^[1792]
- [Generate Source Code](#)^[1768]

10.3.3 C Conventions

Enterprise Architect supports round trip engineering of C, where the following conventions are used:

Reference

Stereotype:

Stereotype	Applies To	Corresponds To
enumeration	Inner Class	An <i>enum</i> type.
struct	Inner Class	A <i>struct</i> type.
	Attribute	A keyword <i>struct</i> in variable definition.
typedef	Inner Class	A <i>typedef</i> statement, where the parent is the original type name.
union	Inner Class	A <i>union</i> type.

Stereotype	Applies To	Corresponds To
	Attribute	A keyword <i>union</i> in variable definition.

Tagged Values:

Tag	Applies To	Corresponds To
anonymous	Class also containing the Tagged Value <i>typedef</i>	The name of this class being defined only by the <i>typedef</i> statement.
bodyLocation	Operation	The location the method body is generated to; expected values are header , classDec or classBody .
typedef	Class with stereotype other than <i>typedef</i>	This Class being defined in a <i>typedef</i> statement.

C Code Generation for UML Model:

UML	C Code	Notes
A Class	A pair of C files (.h + .c)	File name is the same as Class name.
Operation (public & protected)	Function declaration in .h file and definition in .c file	
Operation (private)	Function definition in .c file only	
Attribute (public & protected)	Variable definition in .h file	
Attribute (private)	Variable definition in .c file	
Inner Class (without stereotype)	(N/A)	This inner Class would be ignored

Topics

Topic	Detail	See also
Capture #define value to be generated in C code	For example, <i>#define PI 3.14</i> 1. Add an attribute to the Class, with Name = PI and Initial Value = 3.14 2. In the General page of the Attribute Properties dialog, select the Static and Const checkboxes 3. On the Tagged Values page of the Attribute Properties dialog, add a tag called define with the value true	Attributes ^[885] Attributes Dialog - General ^[887] Attribute Tagged Values ^[891]

Notes

- Separate conventions apply to Object Oriented programming in C

Learn more

- [Import Source Code](#)^[1792]
- [Generate Source Code](#)^[1768]
- [C Options](#)^[1911]
- [Object Oriented Programming in C](#)^[1747]

10.3.3.1 Object Oriented Programming In C

The following conventions are used for Object-Oriented programming in C.

To configure Enterprise Architect to support Object-Oriented programming using C, you must set the **Object Oriented Support** option to **True** on the C Specifications page of the Options dialog.

Reference**Stereotype**

Stereotype	Applies To	Corresponds To
enumeration	Class	An <i>enum</i> type.
struct	Class	A <i>struct</i> type.
	Attribute	A keyword <i>struct</i> in variable definition.
typedef	Class	A <i>typedef</i> statement, where the parent is the original type name.
union	Class	A <i>union</i> type.
	Attribute	A keyword <i>union</i> in variable definition.

Tagged Values

Tag	Applies To	Corresponds To
anonymous	Class with stereotype of <i>enumeration, struct or union</i>	The name of this Class being defined only by the <i>typedef</i> statement.
bodyLocation	Operation	The location the method body is generated to; expected values are header , classDec or classBody .
define	Attribute	<i>#define</i> statement.
typedef	Class with stereotype of <i>enumeration, struct or union</i>	This Class being defined in a <i>typedef</i> statement.

Topics

Topic	Detail	See also
Object-Oriented C Code Generation for UML Model	<p>The basic idea of implementing a UML Class in C code is to group the data variable (UML attributes) into a structure type; this structure is defined in a .h file so that it can be shared by other classes and by the client that referred to it.</p> <p>An operation in a UML Class is implemented in C code as a function; the name of the function must be a fully qualified name that consists of the operation name, as well as the Class name to indicate that the operation is for that Class.</p> <p>A delimiter (specified in the Namespace Delimiter option on the C Specifications page) is used to join the Class name and function (operation) name.</p> <p>The function in C code must also have a reference parameter to the Class object - you can modify the Reference as Operation Parameter, Reference Parameter Style and Reference Parameter Name options on the C Specifications page to support this reference parameter.</p>	<p>C Specification ^[1917]</p>
Limitations of Object-Oriented Programming in C	<ul style="list-style-type: none"> • No scope mapping for an attribute: an attribute in a UML Class is mapped to a structure variable in C code, and its scope (private, protected or public) is ignored • Currently an inner Class is ignored: if a UML Class is the inner Class of another UML Class, it is ignored when generating C code • Initial value is ignored: the initial value of an attribute in a UML Class is ignored in generated C code 	

Learn more

- [Import Source Code](#) ^[1792]
- [Generate Source Code](#) ^[1768]
- [C Options](#) ^[1911]

10.3.4 C# Conventions

Enterprise Architect supports the round trip engineering of C#, where the following conventions are used.

Reference

Stereotypes

Stereotype	Applies To	Corresponds To
enumeration	Class	An <i>enum</i> type.
event	Operation	An event.

Stereotype	Applies To	Corresponds To
extension	Operation	A Class extension method, represented in code by a <i>this</i> parameter in the signature.
indexer	Operation	A property acting as an index for this Class.
partial	Operation	The <i>partial</i> keyword on an operation.
property	Operation	A property possibly containing both read and write code.
struct	Class	A <i>struct</i> type.

Tagged Values

Tag	Applies To	Corresponds To
argumentName	Operation with stereotype <i>extension</i>	The name given to <i>this</i> parameter.
attributeName	Operation with stereotype <i>property</i> or <i>event</i>	The name of the variable behind this property or event.
className	Operation with stereotype <i>extension</i>	The Class that this method is being added to.
const	Attribute	The <i>const</i> keyword.
definition	Operation with stereotype <i>partial</i>	Whether this is the declaration of the method, or the definition.
delegate	Operation	The <i>delegate</i> keyword.
enumType	Operation with stereotype <i>property</i>	The datatype that the property is represented as.
extern	Operation	The <i>extern</i> keyword.
fixed	Attribute	The <i>fixed</i> keyword.
generic	Operation	The generic parameters for this Operation.
genericConstraints	Templated Class or Interface, Operation with tag <i>generic</i>	The constraints on the generic parameters of this type or operation.
Implements	Operation	The name of the method this implements, including the interface name.
ImplementsExplicit	Operation	The presence of the source interface name in this method declaration.
initializer	Operation	A constructor initialization list.
new	Class, Interface, Operation	The <i>new</i> keyword.
override	Operation	The <i>override</i> keyword.
params	Parameter	A parameter list using the <i>params</i> keyword.
partial	Class, Interface	The <i>partial</i> keyword.

Tag	Applies To	Corresponds To
readonly	Operation with stereotype <i>property</i>	This property only defining read code.
sealed	Operation	The <i>sealed</i> keyword.
static	Class	The <i>static</i> keyword.
unsafe	Class, Interface, Operation	The <i>unsafe</i> keyword.
virtual	Operation	The <i>virtual</i> keyword.
writeonly	Operation with stereotype <i>property</i>	This property only defining write code.

Topics

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> Namespaces are generated for each package below a namespace root The <i>Const</i> property of an attribute corresponds to the <i>readonly</i> keyword, while the tag <i>const</i> corresponds to the <i>const</i> keyword The value of <i>inout</i> for the <i>Kind</i> property of a parameter corresponds to the <i>ref</i> keyword The value of <i>out</i> for the <i>Kind</i> property of a parameter corresponds to the <i>out</i> keyword Partial Classes can be modeled as two separate Classes with the <i>partial</i> tag The <i>Is Leaf</i> property of a Class corresponds to the <i>sealed</i> keyword 	Namespaces ^[1777]

Learn more

- [Import Source Code](#) ^[1792]
- [Generate Source Code](#) ^[1768]
- [C# Options](#) ^[1912]

10.3.5 C++ Conventions

Enterprise Architect supports round trip engineering of C++, including the **Managed C++** and **C++/CLI** extensions, where the following conventions are used.

Reference

Stereotypes

Stereotype	Applies To	Corresponds To
enumeration	Class	An <i>enum</i> type.
friend	Operation	The <i>friend</i> keyword.
property get	Operation	A read property.
property set	Operation	A write property.
struct	Class	A <i>struct</i> type.
typedef	Class	A <i>typedef</i> statement, where the parent is the original type name.
union	Class	A <i>union</i> type.

Tagged Values

Tag	Applies To	Corresponds To
afx_msg	Operation	The <i>afx_msg</i> keyword.
anonymous	Class also containing the Tagged Value <i>typedef</i>	The name of this Class being only defined by the <i>typedef</i> statement.
attribute_name	Operation with stereotype <i>property get</i> or <i>property set</i>	The name of the variable behind this property.
bitfield	Attribute	The size, in bits, allowed for storage of this attribute.
bodyLocation	Operation	The location the method body is generated to; expected values are header , classDec or classBody .
callback	Operation	A reference to the CALLBACK macro.
explicit	Operation	The <i>explicit</i> keyword.
initializer	Operation	A constructor initialization list.
inline	Operation	The <i>inline</i> keyword and inline generation of the method body.
mutable	Attribute	The <i>mutable</i> keyword.
throws	Operation	The exceptions that are thrown by this method.
typedef	Class with stereotype other than <i>typedef</i>	This Class being defined in a <i>typedef</i> statement.
typeSynonyms	Class	The <i>typedef</i> name and/or fields of this type.
volatile	Operation	The <i>volatile</i> keyword.

Topics

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> Namespaces are generated for each package below a namespace root <i>By Reference</i> attributes correspond to a pointer to the type specified The <i>Transient</i> property of an attribute corresponds to the <i>volatile</i> keyword The <i>Abstract</i> property of an attribute corresponds to the <i>virtual</i> keyword The <i>Const</i> property of an operation corresponds to the <i>const</i> keyword, specifying a constant return type The <i>Is Query</i> property of an operation corresponds to the <i>const</i> keyword, specifying the method doesn't modify any fields The <i>Pure</i> property of an operation corresponds to a pure <i>virtual</i> method using the "0" syntax The <i>Fixed</i> property of a parameter corresponds to the <i>const</i> keyword 	Namespaces ^[1777]

Learn more

- [Managed C++ Options](#)^[1752]
- [C++/CLI Conventions](#)^[1753]
- [Import Source Code](#)^[1792]
- [Generate Source Code](#)^[1768]
- [C++ Options](#)^[1912]

10.3.5.1 Managed C++ Conventions

The following conventions are used for managed extensions to C++ prior to C++/CLI. In order to set Enterprise Architect to generate managed C++ you must modify the C++ version in the C++ Options.

Reference

Stereotypes

Stereotype	Applies To	Corresponds To
property	Operation	The <code>__property</code> keyword.
property get	Operation	The <code>__property</code> keyword and a read property.
property set	Operation	The <code>__property</code> keyword and a write property.
reference	Class	The <code>__gc</code> keyword.

Stereotype	Applies To	Corresponds To
value	Class	The <code>__value</code> keyword.

Tagged Values

Tag	Applies To	Corresponds To
managedType	Class with stereotype <i>reference</i> , <i>value</i> or <i>enumeration</i> ; Interface	The keyword used in declaration of this type; expected values are <i>class</i> or <i>struct</i> .

Topics

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> The <i>typedef</i> and anonymous tags from native C++ are not supported The <i>Pure</i> property of an operation corresponds to the keyword <code>__abstract</code> 	

Learn more

- [C++/CLI Conventions](#) ^[1753]
- [C++ Options](#) ^[1912]
- [Import Source Code](#) ^[1792]
- [Generate Source Code](#) ^[1768]

10.3.5.2 C++/CLI Conventions

The following conventions are used for modeling C++/CLI extensions to C++. In order to set Enterprise Architect to generate managed C++/CLI you must modify the C++ version in the C++ Options.

Reference

Stereotypes

Stereotype	Applies To	Description
event	Operation	Defines an event to provide access to the event handler for this Class.
property	Operation, Attribute	This is a property possibly containing both read and write code.
reference	Class	Corresponds to the <i>ref class</i> or <i>ref struct</i> keyword.
value	Class	Corresponds to the <i>value class</i> or <i>value struct</i> keyword.

Tagged Values

Tag	Applies To	Description
attribute_name	Operation with stereotype <i>property</i> or <i>event</i>	The name of the variable behind this property or event.
generic	Operation	Defines the generic parameters for this Operation.
genericConstraints	Templated Class or Interface, Operation with tag <i>generic</i>	Defines the constraints on the generic parameters for this Operation.
initonly	Attribute	Corresponds to the <i>initonly</i> keyword.
literal	Attribute	Corresponds to the <i>literal</i> keyword.
managedType	Class with stereotype <i>reference</i> , <i>value</i> or <i>enumeration</i> ; Interface	Corresponds to either the <i>class</i> or <i>struct</i> keyword.

Topics

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> The <i>typedef</i> and anonymous tags are not used The <i>property get/property set</i> stereotypes are not used The <i>Pure</i> property of an operation corresponds to the keyword <i>abstract</i> 	

Learn more

- [C++ Options](#) ^[1912]
- [Import Source Code](#) ^[1792]
- [Generate Source Code](#) ^[1768]

10.3.6 Delphi Conventions

Enterprise Architect supports round trip engineering of Delphi, where the following conventions are used.

Reference

Stereotypes

Stereotype	Applies To	Corresponds To
constructor	Operation	A constructor.

Stereotype	Applies To	Corresponds To
destructor	Operation	A destructor.
dispinterface	Class, Interface	A dispatch interface.
enumeration	Class	An enumerated type.
metaclass	Class	A metaclass type.
object	Class	An object type.
operator	Operation	An operator.
property get	Operation	A read property.
property set	Operation	A write property.
struct	Class	A record type.

Tagged Values

Tag	Applies To	Corresponds To
attribute_name	Operation with stereotype <i>property get</i> or <i>property set</i>	The name of the variable behind this property.
overload	Operation	The <i>overload</i> keyword.
override	Operation	The <i>override</i> keyword.
packed	Class	The <i>packed</i> keyword.
property	Class	A property; see <i>Delphi Properties</i> for more information.
reintroduce	Operation	The <i>reintroduce</i> keyword.

Topics

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> The <i>Static</i> property of an attribute or operation corresponds to the <i>class</i> keyword The <i>Fixed</i> property of a parameter corresponds to the <i>const</i> keyword The value of <i>inout</i> for the <i>Kind</i> property of a parameter corresponds to the <i>Var</i> keyword The value of <i>out</i> for the <i>Kind</i> property of a parameter corresponds to the <i>Out</i> keyword 	

[Learn more](#)

- [Import Source Code](#)^[1792]
- [Generate Source Code](#)^[1768]
- [Delphi Options](#)^[1913]
- [Delphi Properties](#)^[1914]

10.3.7 Java Conventions

Enterprise Architect supports round trip engineering of Java - including **AspectJ** extensions - where the following conventions are used.

Reference

Stereotypes

Stereotype	Applies To	Corresponds To
annotation	Interface	An <i>annotation</i> type.
enum	Attributes within a Class stereotyped <i>enumeration</i>	An <i>enumerated</i> option, distinguished from other attributes that have no stereotype.
enumeration	Class	An <i>enum</i> type.
operator	Operation	An operator.
property get	Operation	A read property.
property set	Operation	A write property.
static	Class or Interface	The <i>static</i> keyword.

Tagged Values

Tag	Applies To	Corresponds To
annotations	Anything	The annotations on the current code feature.
arguments	Attribute with stereotype <i>enum</i>	The arguments that apply to this enumerated value.
attribute_name	Operation with stereotype <i>property get</i> or <i>property set</i>	The name of the variable behind this property.
dynamic	Class or Interface	The <i>dynamic</i> keyword.
generic	Operation	The generic parameters to this operation.
parameterList	Parameter	A parameter list with the ... syntax.
throws	Operation	The exceptions that are thrown by this method.
transient	Attribute	The <i>transient</i> keyword.

Topics

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> Package statements are generated when the current package is not a namespace root The <i>Const</i> property of an attribute or operation corresponds to the final keyword The <i>Transient</i> property of an attribute corresponds to the volatile keyword The <i>Fixed</i> property of a parameter corresponds to the final keyword 	Namespaces <small>[1777]</small>

Learn more

- [AspectJ Conventions](#)
[1757]
- [Import Source Code](#)
[1792]
- [Generate Source Code](#)
[1768]
- [Java Options](#)
[1915]

10.3.7.1 AspectJ Conventions

The following are the conventions used for supporting AspectJ extensions to Java.

Reference**Stereotypes**

Stereotype	Applies To	Corresponds To
advice	Operation	A piece of advice in an AspectJ aspect.
aspect	Class	An AspectJ aspect.
pointcut	Operation	A pointcut in an AspectJ aspect.

Tagged Values

Tag	Applies To	Corresponds To
className	Attribute or operation within a Class stereotyped <i>aspect</i>	The Classes this AspectJ intertype member belongs to.

Topics

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> The specifications of a pointcut are included in the Behavior field of the method. 	

Learn more

- [Import Source Code](#)^[1792]
- [Generate Source Code](#)^[1768]

10.3.8 PHP Conventions

Enterprise Architect supports the round trip engineering of PHP 4 and 5, where the following conventions are used.

Reference

Stereotypes

Stereotype	Applies To	Corresponds To
property get	Operation	A read property.
property set	Operation	A write property.

Tagged Values

Tag	Applies To	Corresponds To
attribute_name	Operation with stereotype <i>property get</i> or <i>property set</i>	The name of the variable behind this property..
final	Operations in PHP 5.	The final keyword.

Topics

Topic	Detail	See also
Common Conventions	<ul style="list-style-type: none"> An unspecified type is modeled as <i>var</i> Methods returning a reference are generated by setting the <i>Return Type</i> to <i>var*</i> Reference parameters are generated from parameters with the parameter <i>Kind</i> set to <i>inout</i> or <i>out</i> 	
PHP 5 Conventions	<ul style="list-style-type: none"> The <i>final</i> Class modifier corresponds to the <i>Is Leaf</i> property The <i>abstract</i> Class modifier corresponds to the 	

Topic	Detail	See also
	<p><i>Abstract</i> property</p> <ul style="list-style-type: none"> Parameter type hinting is supported by setting the <i>Type</i> of a parameter The value of <i>inout</i> or <i>out</i> for the <i>Kind</i> property of a parameter corresponds to a <i>reference</i> parameter 	

Learn more

- [Import Source Code](#) ^[1792]
- [Generate Source Code](#) ^[1768]
- [PHP Options](#) ^[1916]

10.3.9 Python Conventions

Enterprise Architect supports the round trip engineering of Python, where the following conventions are used.

Reference

Tagged Values

Tag	Applies To	Corresponds To
Decorators	Class, Operation	The decorators applied to this element in the source.

Topics

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> Model members with <i>Private Scope</i> correspond to code members with two leading underscores Attributes are only generated when the Initial value is not empty All types are reverse engineered as <i>var</i> 	

Learn more

- [Import Source Code](#) ^[1792]
- [Generate Source Code](#) ^[1768]
- [Python Options](#) ^[1917]

10.3.10 System C Conventions

Enterprise Architect supports round-trip engineering of SystemC, where the following conventions are used.

Reference

Stereotypes

Stereotype	Applies To	Corresponds To
delegate	Method	A delegate.
enumeration	Inner Class	An <i>enum</i> type.
friend	Method	A <i>friend</i> method.
property	Method	A property definition.
sc_ctor	Method	A SystemC constructor.
sc_module	Class	A SystemC module.
sc_port	Attribute	A port.
sc_signal	Attribute	A signal.
struct	Inner Class	A <i>struct</i> or <i>union</i> .

Tagged Values

Tag	Applies To	Corresponds To
kind	Attribute (Port)	Port kind (<i>clocked, fifo, master, slave, resolved, vector</i>).
mode	Attribute (Port)	Port mode (<i>in, out, inout</i>).
overrides	Method	The <i>Inheritance</i> list of a method declaration.
throw	Method	The exception specification of a method.

Topics

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> SystemC also inherits most of the stereotypes and Tagged Values of C++ 	C++ Conventions ^[1750]

SystemC Toolbox Pages:

To access the SystemC pages of the Toolbox, select the **More tools | HDL | SystemC Constructs** menu option. Drag these icons onto a diagram to model a SystemC design.

Reference

Page	Item	Action
SystemC	Module	Define a SystemC Module. An <i>sc_module</i> -stereotyped Class element.
	Enumeration	Define an Enumerated Type. An <i>enumeration</i> -stereotyped Enumeration element.
	Struct	Define a Structure. A <i>struct</i> -stereotyped Class element.
SystemC Features	Port	Define a SystemC Port. An <i>sc_port</i> -stereotyped attribute.
	Signal	Define a SystemC Signal. An <i>sc_signal</i> -stereotyped attribute.
	Constructor	Define a SystemC Constructor. An <i>sc_ctor</i> -stereotyped method.

Learn more

- [Import Source Code](#)^[1792]
- [Generate Source Code](#)^[1768]
- [SystemC Options](#)^[1917]

10.3.11 VB.Net Conventions

Enterprise Architect supports round-trip engineering of Visual Basic.Net, where the following conventions are used. Earlier versions of Visual Basic are supported as a different language.


Reference**Stereotypes**

Stereotype	Applies To	Corresponds To
event	Operation	An event declaration.
import	Operation	An operation to be imported from another library.
module	Class	A module.
operator	Operation	An operator overload definition.
partial	Operation	The <i>partial</i> keyword on an operation.
property	Operation	A property possibly containing both read and write code.

Tagged Values

Tag	Applies To	Corresponds To
Alias	Operation with stereotype <i>import</i>	The alias for this imported operation.
attribute_name	Operation with stereotype <i>property</i>	The name of the variable behind this property.
Charset	Operation with stereotype <i>import</i>	The <i>character set</i> clause for this import - one of the values <i>Ansi</i> , <i>Unicode</i> or <i>Auto</i> .
delegate	Operation	The <i>Delegate</i> keyword.
enumTag	Operation with stereotype <i>property</i>	The datatype that this property is represented as.
Handles	Operation	The <i>handles</i> clause on this operation.
Implements	Operation	The <i>implements</i> clause on this operation.
Lib	Operation with stereotype <i>import</i>	The library this import comes from.
MustOverride	Operation	The <i>MustOverride</i> keyword.
Narrowing	Operation with stereotype <i>operator</i>	The <i>Narrowing</i> keyword.
NotOverrideable	Operation	The <i>NotOverrideable</i> keyword.
Overloads	Operation	The <i>Overloads</i> keyword.
Overrides	Operation	The <i>Overrides</i> keyword.
parameterArray	Parameter	A parameter list using the <i>ParamArray</i> keyword.
partial	Class, Interface	The <i>Partial</i> keyword.
readonly	Operation with stereotype <i>property</i>	This property only defining read code.
shadows	Class, Interface, Operation	The <i>Shadows</i> keyword.
Shared	Attribute	The <i>Shared</i> keyword.
Widening	Operation with stereotype <i>operator</i>	The <i>Widening</i> keyword.
writeonly	Operation with stereotype <i>property</i>	This property only defining write code.

Topics

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> Namespaces are generated for each package below a namespace root The <i>IsLeaf</i> property of a Class corresponds to the <i>NotInheritable</i> keyword The <i>Abstract</i> property of a Class corresponds to the <i>MustInherit</i> keyword 	Namespaces 

Topic	Detail	See also
	<ul style="list-style-type: none"> The <i>Static</i> property of an attribute or operation corresponds to the <i>Shared</i> keyword The <i>Abstract</i> property of an operation corresponds to the <i>MustOverride</i> keyword The value of <i>in</i> for the <i>Kind</i> property of a parameter corresponds to the <i>ByVal</i> keyword The value of <i>inout</i> or <i>out</i> for the <i>Kind</i> property of a parameter corresponds to the <i>ByRef</i> keyword 	

Learn more

- [Visual Basic Conventions](#) ^[1766]
- [Import Source Code](#) ^[1792]
- [Generate Source Code](#) ^[1768]
- [VB.Net Options](#) ^[1918]

10.3.12 Verilog Conventions

Enterprise Architect supports round-trip engineering of Verilog, where the following conventions are used.

Reference

Stereotypes

Stereotype	Applies To	Corresponds To
asynchronous	Method	A concurrent process.
enumeration	Inner Class	An <i>enum</i> type.
initializer	Method	An initializer process.
module	Class	A module.
part	Attribute	A component instantiation.
port	Attribute	A port.
synchronous	Method	A sequential process.

Tagged Values

Tag	Applies To	Corresponds To
kind	Attribute (signal)	The signal kind (such as <i>register</i> , <i>bus</i>).

Tag	Applies To	Corresponds To
mode	Attribute (port)	The port mode (<i>in, out, inout</i>).
Portmap	Attribute (part)	The generic / port map of the component instantiated.
sensitivity	Method	The sensitivity list of a sequential process.
type	Attribute	The range or type value of an attribute.

Verilog Toolbox Pages

To access the **Verilog** pages of the **Toolbox**, select the **More tools | HDL | Verilog Constructs** menu option. Drag these icons onto a diagram to model a Verilog design.

Reference

Page	Item	Action
Verilog	Module	Define a Verilog Module. <i>A module-stereotyped Class element.</i>
	Enumeration	Define an Enumerated Type. <i>An enumeration-stereotyped Class element.</i>
Verilog Features	Port	Define a Verilog Port. <i>A port-stereotyped attribute.</i>
	Part	Define a Verilog component instantiation <i>A part-stereotyped attribute.</i>
	Attribute	Define an attribute.
	Procedure <ul style="list-style-type: none"> • Concurrent • Sequential • Initializer. 	Define a Verilog process: <ul style="list-style-type: none"> • An <i>asynchronous</i>-stereotyped method • A <i>synchronous</i>-stereotyped method • An <i>initializer</i>-stereotyped method

Learn more

- [Import Source Code](#)^[1792]
- [Generate Source Code](#)^[1768]
- [Verilog Options](#)^[1919]

10.3.13 VHDL Conventions

Enterprise Architect supports round-trip engineering of VHDL, where the following conventions are used.

Reference

Stereotypes

Stereotype	Applies To	Corresponds To
architecture	Class	An architecture.
asynchronous	Method	An asynchronous process.
configuration	Method	A configuration.
enumeration	Inner Class	An <i>enum</i> type.
entity	Interface	An entity.
part	Attribute	A component instantiation.
port	Attribute	A port.
signal	Attribute	A signal declaration.
struct	Inner Class	A record definition.
synchronous	Method	A synchronous process.
typedef	Inner Class	A <i>type</i> or <i>sub type</i> definition.

Tagged Values

Tag	Applies To	Corresponds To
isGeneric	Attribute (port)	The port declaration in a generic interface.
isSubType	Inner Class (typedef)	A subtype definition.
kind	Attribute (signal)	The signal kind (such as <i>register</i> , <i>bus</i>).
mode	Attribute (port)	The port mode (<i>in</i> , <i>out</i> , <i>inout</i> , <i>buffer</i> , <i>linkage</i>).
portmap	Attribute (part)	The generic / port map of the component instantiated.
sensitivity	Method (synchronous)	The sensitivity list of a synchronous process.
type	Inner Class (typedef)	The type indication of a type declaration.
typeNameSpace	Attribute (part)	The type namespace of the instantiated component.

VHDL Toolbox Pages:

To access the VHDL pages of the Toolbox, select the **More tools | HDL | VHDL Constructs** menu option. Drag these icons onto a diagram to model a VHDL design.

Reference

Page	Item	Action
VHDL	Architecture	Define an architecture to be associated with a VHDL entity. <i>An architecture-stereotyped Class element.</i>
	Entity	Define a VHDL entity to contain the Port definitions. <i>An entity-stereotyped interface element.</i>
	Enumeration	Define an Enumerated Type. <i>An enumeration-stereotyped enumeration element.</i>
	Struct	Define a VHDL record. <i>A struct-stereotyped Class element.</i>
	Typedef	Define a VHDL type or subtype. <i>A typedef-stereotyped Class element.</i>
VHDL Features	Port	Define a VHDL Port. <i>A port-stereotyped attribute.</i>
	Part	Define a VHDL component instantiation. <i>A part-stereotyped attribute.</i>
	Signal	Define a VHDL signal. <i>A signal-stereotyped attribute.</i>
	Procedure <ul style="list-style-type: none"> • Concurrent • Sequential • Configuration. 	Define a VHDL process: <ul style="list-style-type: none"> • <i>An asynchronous-stereotyped method</i> • <i>A synchronous-stereotyped method</i> • <i>A configuration-stereotyped method</i>

Learn more

- [Import Source Code](#)^[1792]
- [Generate Source Code](#)^[1768]
- [VHDL Options](#)^[1919]

10.3.14 Visual Basic Conventions

Enterprise Architect supports the round trip engineering of Visual Basic 5 and 6, where the following conventions are used.

Visual Basic .Net is supported as a different language.

Reference**Stereotypes**

Stereotype	Applies To	Corresponds To
global	Attribute	The <i>Global</i> keyword.
import	Operation	An operation to be imported from another library.
property get	Operation	A property get.
property set	Operation	A property set.
property let	Operation	A property let.
with events	Attribute	The <i>WithEvents</i> keyword.

Tagged Values

Tag	Applies To	Corresponds To
Alias	Operation with stereotype <i>import</i>	The alias for this imported operation.
attribute_name	Operation with stereotype <i>property get</i> , <i>property set</i> or <i>property let</i>	The name of the variable behind this property.
Lib	Operation with stereotype <i>import</i>	The library this import comes from.
New	Attribute	The <i>New</i> keyword.

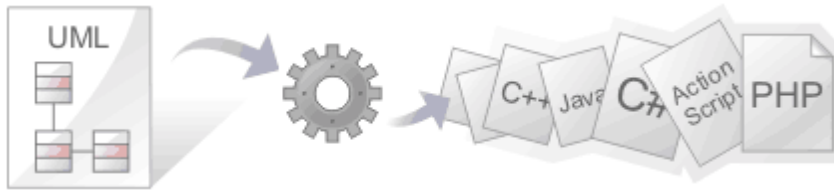
Topics

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> The value of <i>in</i> for the <i>Kind</i> property of a parameter corresponds to the <i>ByVal</i> keyword The value of <i>inout</i> or <i>out</i> for the <i>Kind</i> property of a parameter corresponds to the <i>ByRef</i> keyword 	

Learn more

- [Visual Basic .Net Conventions](#)^[1761]
- [Import Source Code](#)^[1792]
- [Generate Source Code](#)^[1768]
- [Visual Basic Options](#)^[1920]

10.4 Generate Source Code



Generating source code (forward engineering) takes the UML Class or Interface model elements and creates a source code equivalent for future elaboration and compilation. By forward engineering code from the model, the mundane work involved with having to key in Classes and attributes and methods is avoided, and symmetry between model and code is ensured.

Topics

Topic	Detail	See also
Languages	Enterprise Architect supports code generation in each of the following languages: <ul style="list-style-type: none"> • Action Script • C • C# (for both .NET 1.1 and .NET 2.0) • C++ (standard, plus .NET managed C++ extensions) • Delphi • Java (including Java 1.5, Aspects and Generics) • PHP • Python • Visual Basic • Visual Basic .NET 	
Elements	Code is generated from Class or Interface model elements, so you must create the required Class and Interface elements to generate from. Add attributes (which become variables) and operations (which become methods).	
Settings	Before you generate code, you should ensure the default settings for code generation match your requirements; set up the defaults to match your required language and preferences. Preferences that you can define include default constructors and destructors, methods for interfaces and the Unicode options for created languages. Languages such as Java support 'namespaces' and	Source Code Options ^[1892] Namespaces ^[1777] ActionScript Options ^[1742] C Options ^[1911] C# Options ^[1912] C++ Options ^[1912]

	<p>can be configured to specify a namespace root.</p> <p>In addition to the default settings for generating code, Enterprise Architect facilitates setting specific generation options for each of the supported languages.</p>	<p>Delphi Options ^[1913]</p> <p>Java Options ^[1915]</p> <p>PHP Options ^[1916]</p> <p>Python Options ^[1917]</p> <p>Visual Basic Options ^[1920]</p> <p>Visual Basic .NET Options ^[1918]</p>
Code Template Framework	<p>The Code Template Framework (CTF) enables you to customize the way Enterprise Architect generates source code and also enables generation of languages that are not specifically supported by Enterprise Architect.</p>	<p>Code Template Framework (CTF) ^[1924]</p>
Local Paths	<p>Local path names enable you to substitute tags for directory names</p>	<p>Local Paths ^[1900]</p> <p>Local Paths Dialog ^[1901]</p>
Behavioral Code	<p>You can also generate code from three UML behavioral modeling paradigms:</p> <ul style="list-style-type: none"> • State Machine diagrams (SW & HW) • Interaction (Sequence) diagrams (SW) • Activity diagrams (SW) 	<p>State Machine diagrams ^[1780]</p> <p>Interaction (Sequence) diagrams ^[1786]</p> <p>Activity diagrams ^[1787]</p>
Live Code Generation	<p>On the Code Engineering submenu, you have the option to update your source code instantly as you make changes to your model.</p>	<p>Code Engineering Sub-Menu ^[575]</p>
Tasks	<p>When you generate code, you perform one or more of the following tasks:</p> <ul style="list-style-type: none"> • Generate a Single Class • Generate a Group of Classes • Generate a Package • Update Package Contents 	<p>Generate a Single Class ^[1770]</p> <p>Generate a Group of Classes ^[1771]</p> <p>Generate a Package ^[1772]</p> <p>Update Package Contents ^[1774]</p>

Notes

- The tools provided by Enterprise Architect for code engineering and debugging are available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- When security is enabled there are permissions for Generate Source Code and DDL and Reverse Engineer from DDL and Source Code

Learn more

- [Importing Source Code](#)^[1790]
- [Generate From Behavioral Models](#)^[1778]

Learning Center topics

- **(Alt + F1) | Software Engineering | Generate Code**

10.4.1 Generate a Single Class

To generate code for a single Class, first ensure the design of the model element (Class or Interface) is complete. Also ensure you have added Inheritance connectors to parents and associations to other Classes that are used. Also add Inheritance connectors to Interfaces that your Class implements; Enterprise Architect offers the option to generate function stubs for all interface methods that a Class implements. Once the design is satisfactory, follow the procedure below.

How to

To generate code for a single Class

Step	Action	See also
1	Open the diagram containing the Class or Interface for which to generate code.	
2	Right-click on the required Class or Interface and select the Generate Code context menu option, or press (F11). The Generate Code dialog displays, which enables you to control how and where your source code is generated.	
3	On the Path field, click on (...) (Browse) and select a path name for your source code to be generated to.	
4	In the Target Language field, click on the drop-down arrow and select the language to generate; this becomes the permanent option for that Class, so change it back if you are only doing one pass in another language.	
5	Click on the Advanced button. The Object Options dialog displays.	
6	Set any custom options (for this Class alone), then click on the Close button to return to the Generate Code dialog.	

Step	Action	See also
7	In the Import(s) / Header(s) fields, type any import statements, <i>#includes</i> or other header information. Note that in the case of Visual Basic this information is ignored; in the case of Java the two import text boxes are merged; and in the case of C++ the first import text area is placed in the header file and the second in the body (.cpp) file.	
8	Click on the Generate button to create the source code.	
9	When complete, click on the View button to see what has been generated. Note that you should set up your default viewer/editor for each language type first; you can also set up the default editor on the Code Editors page of the Options dialog (Tools Options Source Code Engineering Code Editors).	Editing Source Code <small>[1799]</small>

Learning Center topics

- **(Alt + F1) | Software Engineering | Generate Code | Generate a Single Class**

10.4.2 Generate a Group of Classes

In addition to being able to generate code for an individual Class, you can also select a group of Classes for batch code generation. When you do this, you accept all the default code generation options for each Class in the set.

How to

Step	Detail	See also
1	Select a group of Classes and/or interfaces in a diagram.	
2	Right-click on an element in the group to display the context menu.	
3	Select the Code Generation Generate Selected elements menu option. If no code exists for the selected elements, the Save As dialog displays on which you specify the file path and name for each code file; enter this information and click on the Save button.	
4	The Batch Generation dialog displays, showing the status of the process as it executes (the process might be too fast to see this dialog). If code already exists for the selected Class elements, and changes have been made to the Class name or structure, the Synchronize Element <package name> .<element name> dialog might also display; this dialog helps synchronize the model and code.	Synchronize Model and Code <small>[1778]</small>

--	--	--

Notes

- If any of the elements selected are not Classes or interfaces the option to generate code is not available

Learning Center:

- (Alt + F1) | [Software Engineering | Generate Code | Generate Multiple Classes](#)

10.4.3 Generate a Package

In addition to generating source code from single Classes and groups of Classes, you can also generate code from a package. This feature provides options to recursively generate code from child packages and automatically generate directory structures based on the package hierarchy. This enables you to generate code for a whole branch of your project model in one step.

Access [Project Browser package context menu | Code Engineering | Generate Source Code](#)

How to

To generate code from a package, on the Generate Package Source Code dialog, follow the steps below:

Step	Action	See also
1	<p>In the Synchronize field, click on the drop-down arrow and select the appropriate synchronize option:</p> <ul style="list-style-type: none"> • Synchronize model and code: Code for Classes with existing files is forward synchronized with that file; code for Classes with no existing file is generated to the displayed target file • Overwrite code: All selected target files are overwritten (forward generated) • Do not generate: Generate code for only those selected Classes that do not have an existing file; all other Classes are ignored 	
2	<p>Highlight the Classes for which to generate code; leave unselected any to not generate code for.</p> <p>If you want to display more of the information within the layout, you can resize the dialog and its columns.</p>	
3	<p>To make Enterprise Architect automatically generate directories and filenames based on the package hierarchy, select the Auto Generate Files checkbox; this enables the Root Directory field, in which you select a root directory under which the source directories are to be generated.</p> <p>By default, the Auto Generate Files feature <i>ignores</i> any file paths that are already associated with a Class; you can change this behavior by also selecting the Retain Existing File Paths checkbox.</p>	

Step	Action	See also
4	To include code for all sub-packages in the output, select the Include Child Packages checkbox.	
5	Click on the Generate button to start generating code.	

As code generation proceeds Enterprise Architect displays progress messages. If a Class requires an output filename Enterprise Architect prompts you to enter one at the appropriate time (assuming **Auto Generate Files** is not selected). For example, if the selected Classes include partial Classes, a prompt displays to enter the filename into which to generate code for the second partial Class.

For additional information on the options on the Generate Package Source Code dialog, see the following table:

Option	Action	See also
Root Package	Check the name of the package for which code is to be generated.	
Synchronize	Select options that specify how existing files should be regenerated.	
Auto Generate Files	Specify whether Enterprise Architect should automatically generate file names and directories, based on the package hierarchy.	
Root Directory	If Auto Generate Files is selected, display the path under which the generated directory structures are created.	
Retain Existing File Paths	If Auto Generate Files is selected, specify whether to use existing file paths associated with Classes. If Auto Generate Files is unselected, Enterprise Architect generates Class code to automatically determined paths, regardless of whether source files are already associated with the Classes.	
Include all Child Packages	Also generate code for all Classes in all sub-packages of the target package in the list. This option facilitates recursive generation of code for a given package and its sub-packages.	
Select Objects to Generate	List all Classes that are available for code generation under the target packages; only code for selected (highlighted) Classes is generated.	

Option	Action	See also
	Classes are listed with their target source file.	
Select All	Mark all Classes in the list as selected.	
Select None	Mark all Classes in the list as unselected.	
Generate	Start the generation of code for all selected Classes.	
Cancel	Exit the Generate Package Source Code dialog; no Class code is generated.	

Learning Center topics

- (Alt + F1) | [Software Engineering | Generate Code | Generate a Package](#)

10.4.4 Update Package Contents

In addition to generating and importing code, Enterprise Architect provides the option to synchronize the model and source code, creating a model that represents the latest changes in the source code and vice versa. You can use either the model as the source, or the code as the source.

The behavior and actions of synchronization depend on the settings you have selected on the Attributes and Operations page of the Options dialog. These settings enable you to either protect or automatically discard information in the model that is not present in the code, and to prompt for a decision on code features that are not in the model. In the following two example, the appropriate checkboxes have been selected for maximum protection of data:

- You generated some source code, but made subsequent changes to the model. When you generate code again, Enterprise Architect *adds* any new attributes or methods to the existing source code, leaving intact what already exists. This means developers can work on the source code and then generate *additional* methods as required from the model, without having their code overwritten or destroyed.
- You might have made changes to a source code file, but the model has detailed notes and characteristics you do not want to lose. By synchronizing from the source code into the model, you import additional attributes and methods but do not change other model elements.

Using the synchronization methods, it is simple to keep source code and model elements up to date and synchronized.

Access [Project Browser package context menu | Code Engineering | Synchronize Package With Code](#)

Reference

On the Synchronize Package Contents dialog, synchronize the selected model package tree of Classes and the source code for those Classes, as indicated below:

Field/Button	Action	See also
Update Type	Select the radio button to either Forward Engineer or Reverse Engineer the package Classes, as appropriate.	
Include child packages in generation	Select the checkbox to include child packages in the synchronization.	
OK	<p>Click on the button to start synchronization.</p> <p>Enterprise Architect uses the directory names specified when the project source was first imported/generated and updates either the model or the source code depending on the option chosen. IF:</p> <ul style="list-style-type: none"> • Performing forward synchronization AND • There are differences between the model and code AND • The On forward synch, prompt to delete code features not in model checkbox is selected in the Options - Attributes and Operations dialog <p>THEN the Synchronize Element <package name>.<element name> dialog displays.</p> <p>Otherwise, no further action is required.</p>	<p>Synchronize Model and Code [1778]</p>

Notes

- Code synchronization does not change method bodies; behavioral code cannot be synchronized, and code generation only works when generating the entire file
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Generate Source Code and DDL** permission to synchronize source code with model elements

Learn more

- [Options - Attributes and Operations](#) [1899]
- [Configure User Security](#) [251]
- [Generate From Behavioral Models](#) [1778]

Learning Center topics

- (Alt + F1) | **Software Engineering | Generate Code | Update Packages**

10.4.4.1 Synchronize Model and Code

You might either:

- Synchronize the code for a package of Classes against the model in the Project Browser, or
- Regenerate code from a batch of Classes in the model

In such processes, there might be items in the code that are not present in the model.

If you want to trap those items and resolve them manually, select the **On forward synch, prompt to delete code features not in model** checkbox in the Options - Attributes and Operations dialog, so that the Synchronize Element <package name> .<element name> dialog displays, providing options to respond to each item.

Reference

On the Synchronize Element <package name> .<element name> dialog, select one or more items and process them as indicated below:


Button	Detail	See also
Select All	Highlight and select all items in the Feature column.	
Clear All	Deselect and remove highlighting from all items in the Feature column.	
Delete	Mark the selected code features to be removed from the code (the value in the Action column changes to Delete).	
Reassign	<p>Mark the selected code features to be reassigned to elements in the model.</p> <p>This is only possible when an appropriate model element is present that is not already defined in the code.</p> <p>The Select the Corresponding Class Feature dialog displays, from which you select the Class to reassign the feature to. Click on the OK button to mark the feature for reassignment.</p>	
Ignore	Mark the selected code elements not present in the model to be ignored completely (the default; the value in the Action column remains as or changes to <none>).	
Reset to Default	Reset the selected items to Ignore (the value in the Action column changes to <none>).	
OK	Make the assigned changes to the items, and close the dialog.	

[Learn more](#)

- [Update Package Contents](#)^[1774]
- [Generate a Group of Classes](#)^[1771]
- [Options - Attributes and Operations](#)^[1895]

10.4.5 Namespaces

Topics

Topic	Detail	See also
Usage	<p>Languages such as Java support package structures or namespaces.</p> <p>Enterprise Architect lets you specify a package as a namespace root, which denotes where the namespace structure starts; all subordinate packages below this point are generated as namespaces to code.</p> <p>To define a package as a namespace root, right-click on the package in the Project Browser and select the Code Engineering Set as Namespace Root context menu option; the package icon in the Project Browser changes to include a colored corner ().</p> <p>When you have set the namespace root, the menu option changes to Clear Namespace Root; click on this option to take the namespace root status off the package (also, see the context menu described below).</p> <p>Once you have set a namespace root, Java code generated beneath this root automatically adds a package declaration at the head of the generated file indicating the current package location.</p> <p>To view a list of namespaces, select the Settings Namespace Roots menu option; the Namespaces dialog displays.</p> <p>If you double-click on a namespace in the list, the package is highlighted in the Project Browser; alternatively, right-click on the namespace to display a context menu, and select the Locate Package in Browser menu option.</p> <p>You can also clear the selected namespace, by selecting the Clear Namespace Attribute option.</p>	

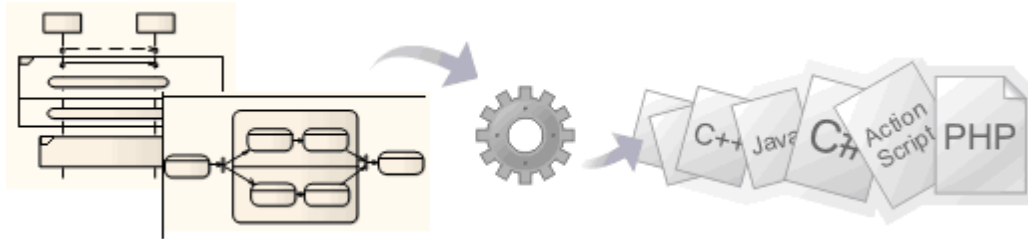
Notes

- When performing code generation, any package name that contains whitespace characters is automatically treated as a namespace root

Learning Center topics

- **(Alt + F1) | Software Engineering | Generate Code | Namespaces**

10.5 Generate From Behavioral Models



Enterprise Architect's system engineering capability supports code generation from the following UML behavioral diagrams:

Topic	Link
State Machine diagrams(SW & HW).	State Machine diagrams ^[1780]
Interaction (Sequence) diagrams (SW).	Interaction (Sequence) diagrams ^[1786]
Activity diagrams (SW).	Activity diagrams ^[1787]

You can generate code in various **software** (SW) and **hardware** (HW) languages, including C(OO), C++, C#, Java, VB.Net, VHDL, Verilog and SystemC.

How to

To experiment with code generation from these diagrams, using the EAExample project provided with your Enterprise Architect installer

Step	Action	See also
1	Open the <i>EAExample.eap</i> file by selecting the Help Open Example Model menu option.	
2	<p>From the Project Browser, select any of the following packages:</p> <p>Software Language Examples:</p> <ul style="list-style-type: none"> Project Models > System Model > Implementation Model (PSM) > Java Model With Behaviors Generate the Account and Order classes Project Models > Systems Engineering Model > Implementation Model > Software > C# Generate the DataProcessor Class Project Models > Systems Engineering Model > Implementation Model > Software > C++ Generate the IO Class Project Models > Systems Engineering Model > Implementation Model > Software > Java Generate the IO Class 	

Step	Action	See also
	<ul style="list-style-type: none"> • Project Models > Systems Engineering Model > Implementation Model > Software > VNet Generate the IO Class <p>Hardware Language Examples:</p> <ul style="list-style-type: none"> • Project Models > Systems Engineering Model > Implementation Model > Hardware > SystemC Generate the PlayBack Class • Project Models > Systems Engineering Model > Implementation Model > Hardware > VHDL Generate the PlayBack Class • Project Models > Systems Engineering Model > Implementation Model > Hardware > Verilog Generate the PlayBack Class 	
3	<p>When completed, press (Ctrl+E) to open the generated source code.</p> <p>You should see methods generated in the code.</p>	

Notes

- Software code generation from behavioral models is available in the Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect
- Hardware code generation from State Machine models is available in the Systems Engineering and Ultimate editions of Enterprise Architect
- For C(OO), please ensure that, on the C Specifications page of the Options dialog, you have set the **Object Oriented Support** option to **True**
- To be able to generate code from behavioral models, all behavioral constructs should be contained within a Class
- Code synchronization is not supported for behavioral code

Learn more

- [State Machine modeling for HDLs](#)^[1784]
- [Behavior](#)^[899]

Learning Center topics

- (Alt + F1) | **Software Engineering | Behavioral Code Generation**

10.5.1 SW Code Generation - State Machine Diagrams

A State Machine in a *Class* internally generates the following constructs in software languages to enable effective execution of the States' behaviors (**do**, **entry** and **exit**) and also to code the appropriate transition's effect when necessary.

Topics

Topic	Detail	See also
Enumerations	<ul style="list-style-type: none"> • StateType - comprises an enumeration for each of the States contained within the State Machine • TransitionType – comprises an enumeration for each transition that has a valid effect associated with it; for example ProcessOrder_Delivered_to_ProcessOrder_Closed • CommandType – comprises an enumeration for each of the behavior types that a State can contain (Do, Entry, Exit) 	State Machine Diagrams ^[1032] Java Code Generated From State Machine Diagram ^[1781]
Attributes	<ul style="list-style-type: none"> • currState:StateType - a variable to hold the current State's information • nextState:StateType - a variable to hold the next State's information, set by each State's transitions accordingly • currTransition:TransitionType - a variable to hold the current transition information; this is set if the transition has a valid effect associated with it • transcend:Boolean - a flag used to advise if a transition is involved in transcending between different State Machines (or Submachine states) • xx_history:StateType - a history variable for each State Machine/Submachine State, to hold information about the last State from which the transition took place 	State Machine Diagrams ^[1032] Java Code Generated From State Machine Diagram ^[1781]
Operations	<ul style="list-style-type: none"> • StatesProc - a States procedure, containing a map between a State's enumeration and its operation; it de-references the current State's information to invoke the respective State's function • TransitionsProc - a Transitions procedure, containing a map between the Transition's enumeration and its effect; it invokes the respective effect • <<State>> - an operation for each of the States contained within the State Machine; this renders a State's behaviors based on the input CommandType, and also executes its transitions • initializeStateMachine - a function that initializes all the framework-related attributes • runStateMachine - a function that iterates through each State, and executes their behaviors and transitions accordingly 	State Machine Diagrams ^[1032] Java Code Generated From State Machine Diagram ^[1781] Operations Dialog - Behavior ^[904]

Notes

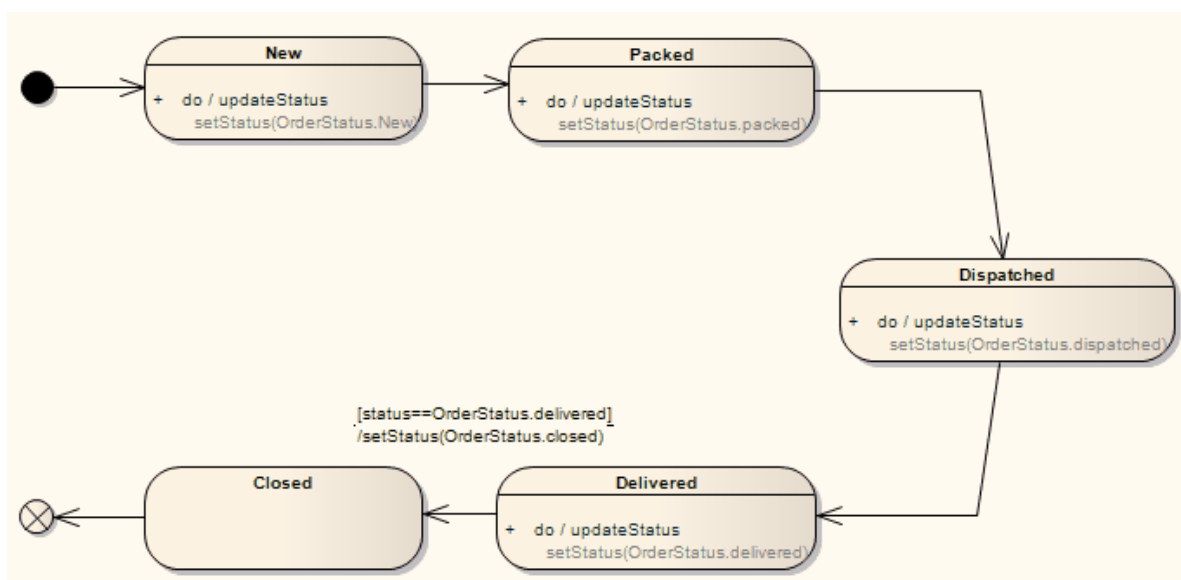
- To be able to generate code from behavioral models, all behavioral constructs should be contained within a Class

Learn more

- [State Machine Diagrams](#) ^[1032]
- [Generate From Behavioral Models](#) ^[1778]
- [State Machine Modeling For HDLs](#) ^[1784]

Learning Center topics

- (Alt + F1) | **Software Engineering | Behavioral Code Generation | State Machines**

10.5.1.1 Java Code Generated From State Machine Diagram

```

private enum StateType : int
{
    ProcessOrder_Delivered,
    ProcessOrder_Packed,
    ProcessOrder_Closed,
    ProcessOrder_Dispatched,
    ProcessOrder_New,
    ST_NOSTATE
}
private enum TransitionType : int
{
    ProcessOrder_Delivered_to_ProcessOrder_Closed,
    TT_NOTRANSITION
}
private enum CommandType
{
    Do,
    Entry,
  
```

```
        Exit
    }
    private StateType currState;
    private StateType nextState;
    private TransitionType currTransition;
    private boolean transcend;
    private StateType ProcessOrder_history;
    private void processOrder_Delivered(CommandType command)
    {
        switch(command)
        {
            case Do:
            {
                // Do Behaviors..
                set Status(Delivered);
                // State's Transitions
                if((status==Delivered))
                {
                    nextState = StateType.ProcessOrder_Closed;
                    currTransition = TransitionType.
ProcessOrder_Delivered_to_ProcessOrder_Closed;
                }
                break;
            }
            default:
            {
                break;
            }
        }
    }

    private void processOrder_Packed(CommandType command)
    {
        switch(command)
        {
            case Do:
            {
                // Do Behaviors..
                set Status(Packed);
                // State's Transitions
                nextState = StateType.ProcessOrder_Dispatched;
                break;
            }
            default:
            {
                break;
            }
        }
    }

    private void processOrder_Closed(CommandType command)
    {
        switch(command)
        {
            case Do:
            {
                // Do Behaviors..
                // State's Transitions
                break;
            }
            default:
            {
                break;
            }
        }
    }

    private void processOrder_Dispatched(CommandType command)
    {
        switch(command)
        {
```

```

        case Do:
        {
            // Do Behaviors..
            set Status(Di spat ched);
            // State's Transitions
            next State = StateType.ProcessOrder_Del i ver ed;
            break;
        }
        default:
        {
            break;
        }
    }
}

private void processOrder_New(CommandType command)
{
    switch(command)
    {
        case Do:
        {
            // Do Behaviors..
            set Status(new);
            // State's Transitions
            next State = StateType.ProcessOrder_Packed;
            break;
        }
        default:
        {
            break;
        }
    }
}

private void StatesProc(StateType currState, CommandType command)
{
    switch(currState)
    {
        case ProcessOrder_Del i ver ed:
        {
            processOrder_Del i ver ed(command);
            break;
        }

        case ProcessOrder_Packed:
        {
            processOrder_Packed(command);
            break;
        }

        case ProcessOrder_Cl osed:
        {
            processOrder_Cl osed(command);
            break;
        }

        case ProcessOrder_Di spat ched:
        {
            processOrder_Di spat ched(command);
            break;
        }

        case ProcessOrder_New:
        {
            processOrder_New(command);
            break;
        }

        default:
            break;
    }
}

private void TransitionsProc(TransitionType transition)

```

```

    {
        switch(transition)
        {
            case ProcessOrder_Delivered_to_ProcessOrder_Closed:
            {
                setStatus(closed);
                break;
            }
            default:
                break;
        }
    }
private void initializeStateMachine()
{
    currState = StateType.ProcessOrder_New;
    nextState = StateType.ST_NOSTATE;
    currTransition = TransitionType.TT_NOTRANSITION;
}

private void runStateMachine()
{
    while(true)
    {
        if ( currState == StateType.ST_NOSTATE )
        {
            break ;
        }

        currTransition = TransitionType.TT_NOTRANSITION;
        StatesProc(currState, CommandType.Do);
// then check if there is any valid transition assigned after the do
behavior
        if ( nextState == StateType.ST_NOSTATE)
        {
            break;
        }

        if ( currTransition != TransitionType.TT_NOTRANSITION )
        {
            TransitionsProc( currTransition );
        }
        if ( currState != nextState)
        {
            StatesProc(currState, CommandType.Exit);
            StatesProc(nextState, CommandType.Entry);
            currState = nextState ;
        }
    }
}

```

10.5.2 State Machine Modeling For HDLs

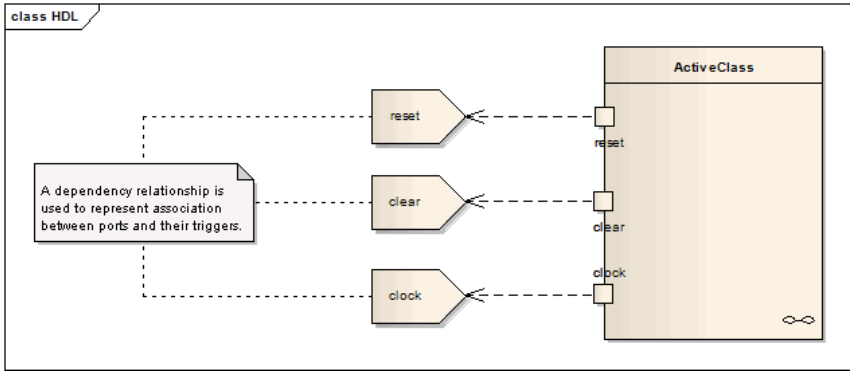
For efficient code generation from State Machine models into Hardware Description Languages (HDL) such as VHDL, Verilog and Systems C, apply the design practices outlined in this topic.

In an HDL State Machine model, the following are expected:

- Designate Driving Triggers
- Establish Port–Trigger Mapping
- Active State Logic

How to

Topic	Detail	See also
-------	--------	----------

<p>Designate Driving Triggers</p>	<ul style="list-style-type: none"> A 'change' trigger is deemed as an asynchronous trigger if the following two conditions are satisfied: <ol style="list-style-type: none"> There is a transition from the actual submachine state (which encapsulates the actual logic) triggered by it The target state of that transition has a self transition triggered by the same trigger Asynchronous triggers should be modeled according to the following pattern: <ol style="list-style-type: none"> The trigger should be of type Change (specification: true / false) The active state (Submachine State) should have a transition trigger by it. The target state of the triggered transition should have a self transition with the same trigger A trigger of type <i>time</i>, which triggers the transitions to the active state (Submachine State) is deemed as the <i>Clock</i>; the specification of this trigger should be specific to the target language: <table border="1" data-bbox="435 896 1297 1176"> <thead> <tr> <th rowspan="2">Trigger Type</th> <th rowspan="2">Language</th> <th colspan="2">Specification</th> </tr> <tr> <th>Positive Edge Triggered</th> <th>Negative Edge</th> </tr> </thead> <tbody> <tr> <td rowspan="3">Time</td> <td>VHDL</td> <td>rising_edge</td> <td>falling_edge</td> </tr> <tr> <td>Verilog</td> <td>posedge</td> <td>negedge</td> </tr> <tr> <td>SystemC</td> <td>positive</td> <td>negative</td> </tr> </tbody> </table>	Trigger Type	Language	Specification		Positive Edge Triggered	Negative Edge	Time	VHDL	rising_edge	falling_edge	Verilog	posedge	negedge	SystemC	positive	negative	<p>State Machine Diagrams [1032]</p> <p>Trigger [1173]</p>
Trigger Type	Language			Specification														
		Positive Edge Triggered	Negative Edge															
Time	VHDL	rising_edge	falling_edge															
	Verilog	posedge	negedge															
	SystemC	positive	negative															
<p>Establish Port-Trigger Mapping</p>	<p>After successfully modeling the different operating modes of the component, and the triggers associated with them, you must associate the triggers with the component's ports</p> <p>A Dependency relationship from the Port to the associated trigger is used to signify their association</p> 																	
<p>Active State Logic</p>	<p>The first two aspects, above, put in place the preliminaries required for efficient interpretation of the hardware components.</p>																	

	The actual State Machine logic is now modeled within the Active (Submachine) state.	
--	---	--

Notes

- To be able to generate code from behavioral models, all behavioral constructs should be contained within a Class
- The current code generation engine supports only one clock trigger for a component

Learn more

- [SW Code Generation - State Machine Diagrams](#) ^[1780]
- [Transition](#) ^[1260]

10.5.3 Code Generation - Interaction Diagrams

During code generation from interaction(sequence) diagrams in a Class, Enterprise Architect applies its system engineering graph optimizer to transform the Class constructs into programmatic paradigms. Messages and Fragments are identified as one of the several action types based on their functionality, and Enterprise Architect uses the *code generation templates* to render their behavior accordingly.

Topics

Topic	Detail	See also
Action Call	A Message that invokes an operation.	Message (Sequence Diagram) ^[1232] Call ^[1236] Message Examples ^[1237]
Action Create	A Message with <i>Lifecycle = New</i> .	Message (Sequence Diagram) ^[1232] Message Properties ^[1233]
Action Destroy	A Message with Lifecycle = Delete.	Message (Sequence Diagram) ^[1232] Message Properties ^[1233]
Action Loop	A Combined Fragment with Type = Alt.	Combined Fragment ^[1110] Interaction Operators ^[1113]
Action If	A Combined Fragment with Type = loop.	Combined Fragment ^[1110] Interaction Operators ^[1113]

Assign To	A call message with a valid target attribute set using the Assign To field is rendered in the code as the target attribute of a call action.	Call ^[1236] Message Properties ^[1233]

Notes

- To be able to generate code from behavioral models, all behavioral constructs should be contained within a Class
- For an Interaction (Sequence) diagram, the behavioral code generation engine expects the Sequence diagram and all its associated messages and interaction fragments to be encapsulated within an interaction element

Learn more

- [EASL Code Generation Macros](#) ^[1449]
- [Interaction](#) ^[1137]
- [Interaction Occurrence](#) ^[1139]
- [Activity](#) ^[1101]

Learning Center topics

- (Alt + F1) | [Software Engineering | Behavioral Code Generation | Interaction Diagrams](#)

10.5.4 Code Generation - Activity Diagrams

Code generation from an Activity Diagrams in a Class requires a validation phase, during which Enterprise Architect uses the system engineering graph optimizer to analyze the diagram and render it into various code-generatable constructs. Enterprise Architect also transforms the constructs into one of the various action types (if appropriate), similar to the Interaction diagram constructs.

Topics

Topic	Detail	See also
Call Actions (Invocation Actions)	Used to invoke operations or behaviors in an Activity diagram; the two main variants of Call Actions supported in behavioral code generation are: <ul style="list-style-type: none"> • CallOperationAction - used to invoke operations, which can be within the same Class or in other Classes within the same package; if referencing operations from other Classes within the same package, you must have a target to which the request is passed • CallBehaviorAction - used to invoke another 	Assign Action Pins ^[1098] Behavior Calls ^[910] Synchronize Arguments ^[911]

	<p>Activity in an activity flow; the referenced activity is expected to be within the same Class</p> <p>Arguments</p> <p>Call Actions can specify argument values corresponding to the parameters in the associated behavior or behavioral feature.</p> <p>You can add the arguments manually or create them automatically using the the Synchronize button of the Arguments dialog.</p>	
CreateObjectAction	<p>Used to denote an object creation in the activity flow; you can set the result Pin of the CreateObjectAction as the object to be created, using the Assign Action Pins dialog.</p> <p>The Classifier of the CreateObjectAction signifies the Classifier for which an instance is to be created.</p>	Assign Action Pins ^[1098]
DestroyObjectAction	<p>Used to denote an object deletion in the activity flow; you can set the target Pin of the DestroyObjectAction as the object to be destroyed, using the Assign Action Pins dialog.</p>	Assign Action Pins ^[1098]
Loops	<p>Enterprise Architect's system engineering graph optimizer is also capable of analyzing and identifying loops; an identified loop is internally rendered as an Action Loop, which is translated by the EASL code generation macros to generate the required code.</p>	
Conditional Statements	<p>To model a conditional statement, you use Decision/Merge nodes.</p> <p>Alternatively, you can imply Decisions/Merges internally; the graph optimizer expects an associated Merge node for each Decision node, to facilitate efficient tracking of various branches and analysis of the code constructs within them.</p>	

Notes

- To be able to generate code from behavioral models, all behavioral constructs should be contained within a Class

Learn more

- [EASL Code Generation Macros](#) ^[1449]
- [Assign Action Pins](#) ^[1098]

- [Interactions and Activities](#)^[910]
- [Behavior Calls](#)^[910]
- [Synchronize Arguments](#)^[911]

Learning Center topics

- (Alt + F1) | **Software Engineering | Behavioral Code Generation | Activity Diagrams**

10.6 Importing Source Code



Topics







Topic	Detail	See also
Usage	<p>Reverse Engineering in Enterprise Architect enables you to import existing source code as a single directory or a directory structure, in a variety of code languages, into a UML model; it helps you to examine, for example, the functionality of code libraries for reuse, or to bring the UML model up to date with new code from outside the model, as part of a process called <i>synchronization</i>.</p> <p>The code structures are mapped into their UML representations: for example, a Java Class is mapped into a UML Class element, with the variables being defined as attributes, methods modeled as operations, and interactions between the Java Classes represented by the appropriate connectors.</p> <p>Examining the code in a UML model helps you to identify the critical modules containing the code, providing a starting point for understanding the business and system requirements of the existing application and enabling the developers to gain a better overall understanding of the source code.</p> <p>Several options are available for performing reverse engineering, including:</p> <ul style="list-style-type: none"> • Whether comments are reverse engineered into notes fields, and how they are formatted if they are • How property methods are recognized • Whether dependencies should be created for operation return and parameter types <p>For further information, see the <i>Source Code Options</i> topic.</p> <p>It is important to note that if a system is not well designed, simply importing the source into Enterprise Architect does not turn it into an easily understandable UML model.</p> <p>When working with a poorly designed system it is useful to assess the code in manageable units by examining the individual model packages or elements generated from the code; for example, dragging a specific Class of interest onto a diagram and then using the Insert Related Elements option at one level to determine the Class's immediate relationship to other Classes.</p> <p>From this point it is possible to create Use Cases that identify the interaction between the source code Classes, enabling an overview of the application's operation.</p>	<p>Import Source Code ^[1792]</p> <p>Import a Directory Structure ^[1795]</p> <p>Source Code Options ^[892]</p> <p>Insert Related Elements ^[833]</p>

Copyright Ownership	<p>Situations that typically lend themselves to reverse engineering tend to operate on source code that:</p> <ul style="list-style-type: none"> • You have already developed • Is part of a third-party library that you have obtained permission to use • Is part of a framework that your organization uses • Is being developed on a daily basis by your developers <p>If you are examining code that you or your organization do not own or do not have specific permission to copy and edit, you must ensure that you understand and comply with the copyright restrictions on that code before beginning the process of reverse engineering.</p>	
----------------------------	---	--

Reference

Enterprise Architect currently supports reverse engineering in the following programming languages:

Topic	Link
Action Script	ActionScript ^[1742]
Ada 2005 (Systems Engineering and Ultimate editions)	Ada 2005 ^[1743]
C	C ^[1745]
C#	C# ^[1748]
C++	C++ ^[1750]
CORBA IDL (MDG Technology)	www.sparxsystems.com/resources/mdg_tech/
Delphi	Delphi ^[1754]
Java	Java ^[1756]
PHP	PHP ^[1758]

Python	Python 
SystemC(Systems Engineering and Ultimate editions)	SystemC 
Verilog (Systems Engineering and Ultimate editions)	Verilog 
VHDL(Systems Engineering and Ultimate editions)	VHDL 
Visual Basic	Visual Basic 
Visual Basic .NET	Visual Basic .NET 

Enterprise Architect can also reverse engineer certain types of binary files - Java .jar files and .NET PE files; see *Import Binary Module* for more information.

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Reverse Engineer From DDL And Source Code** permission to reverse engineer source code and synchronize model elements against code
- Reverse Engineering of other languages is currently available through the use of MDG Technologies from www.sparxsystems.com/resources/mdg_tech/

Learn more

- [Import Binary Module](#) 
- [Permission List](#) 
- www.sparxsystems.com/resources/mdg_tech/ (Online Resource)

Learning Center topics

- (Alt + F1) | **Software Engineering | Import Code**

10.6.1 Import Source Code

How to

To import source code (reverse engineer) follow the steps below:

Step	Action	See also
1	In the Project Browser, select (or add) a diagram into which to import the Classes.	
2	Right-click on the diagram background to open the context menu and either: <ul style="list-style-type: none"> • Select the language to import from the Import from source file(s) submenu • Click on the Import Language drop-down arrow in the Code Generation toolbar and select the Import Import xxx files menu option, where xxx represents the language to import 	
3	From the file browser that appears, select one or more source code files to import.	Notes on Source Code Import ^[1793]
4	Click on the Open button to start the import process.	

As the import proceeds, Enterprise Architect provides progress information. When all files are imported, Enterprise Architect makes a second pass to resolve associations and inheritance relationships between the imported Classes.

Learning Center topics

- (Alt + F1) | **Software Engineering | Import Source Code**

10.6.2 Notes on Source Code Import

Enterprise Architect enables you to import code into your project, in a range of programming languages.

Topics

Topic	Detail	See also
Introduction	Enterprise Architect supports most constructs and keywords for each coding language. If there is a particular feature you require support for that you feel is missing, please contact Sparx Systems. You must select the appropriate type of source file for the language, as the source code to import.	Sparx Systems (Online resource)
ActionScript	Appropriate type of source file: .as code file.	ActionScript ^[1742]
C	Appropriate type of source file: .h header files and/or .c files.	C ^[1745]

	<p>When you select a header file, Enterprise Architect automatically searches for the corresponding .c implementation file to import, based on the options for extension and search path specified in the C options.</p> <p>Enterprise Architect does not expand macros that have been used, these must be added into the internal list of Language Macros.</p>	C options ^[1911] Language Macros ^[1902]
C++	<p>Appropriate type of source file: .h header file.</p> <p>Enterprise Architect automatically searches for the .cpp implementation file based on the extension and search path set in the C++ options; when it finds the implementation file, it can use it to resolve parameter names and method notes as necessary.</p> <p>When importing C++ source code, Enterprise Architect ignores function pointer declarations.</p> <p>To import them into your model you could create a <i>typedef</i> to define a function pointer type, then declare function pointers using that type; function pointers declared in this way are imported as attributes of the function pointer type.</p> <p>Enterprise Architect does not expand macros that have been used; these must be added into the internal list of Language Macros.</p>	C++ ^[1750] C++ options ^[1912] Language Macros ^[1902]
C#	<p>Appropriate type of source file: .cs.</p>	C# ^[1748] C# Options ^[1912]
Delphi	<p>Appropriate type of source file: .pas.</p>	Delphi ^[1754]
Java	<p>Appropriate type of source file: .java.</p> <p>Enterprise Architect supports the AspectJ language extensions.</p> <div data-bbox="480 1469 1007 1906" data-label="Diagram"> <pre> classDiagram class ThingObserving { <<aspect>> -observers: Vector = new Vector() +addObserver(Thing, Thing) : void +removeObserver(Thing, ThingObserver) : void ~updateObserver(Thing, ThingObserver) : void <<advice>> +after(Thing) : void changes(t) <<pointcut>> ~changes(Thing) : void target(t) && call(Void Thing.set*(int)) } </pre> </div> <p>Aspects are modeled using Classes with the stereotype aspect; these aspects can then contain attributes and methods as for a normal Class.</p>	Java ^[1756] Java Options ^[1915]

	<p>If an <i>intertype</i> attribute or operation is required, you can add a tag <i>className</i> with the value being the name of the Class it belongs to.</p> <p><i>Pointcuts</i> are defined as operations with the stereotype <i>pointcut</i>, and can occur in any Java Class, Interface or aspect; the details of the <i>pointcut</i> are included in the behavior field of the method.</p> <p><i>Advice</i> is defined as an operation with the stereotype <i>advice</i>; the <i>pointcut</i> this advice operates on is in the behavior field and acts as part of the method's unique signature.</p> <p><i>afterAdvice</i> can also have one of the Tagged Values returning or throwing.</p>	
PHP	<p>Appropriate type of source file: .php, .php4, or .inc.</p> <p>Nested if condition syntax is enabled.</p>	PHP ^[1758]
Python	<p>Appropriate type of source file: .py.</p>	Python ^[1759]
Visual Basic	<p>Appropriate type of source file: .cls Class file.</p>	Visual Basic ^[1766]
Visual Basic .NET	<p>Appropriate type of source file: .vb Class file.</p>	Visual Basic .NET ^[1761]

Notes

- When reverse engineering attributes with parameter substitutions (templated attributes):
 - If a Class with proper template parameter definitions is found, an Association connector is created and its parameter substitutions are configured
 - An Association connector is also created if a matching entry is defined as a Collection Class or in the **Additional Collection Classes** option (for C#, C++ and Java); for an example, see *Example Use of Collection Classes*

Learn more

- [Import Source Code](#) ^[1792]
- [Parameterized Classes \(Templates\)](#) ^[1183]
- [Set Collection Classes](#) ^[1904]
- [Example Use of Collection Classes](#) ^[1906]

10.6.3 Import a Directory Structure

You can import from all source files in a complete directory structure, which enables you to import or synchronize multiple files in a directory tree in one pass.

Enterprise Architect creates the necessary packages and diagrams during the import process.

Access [Project Browser package context menu | Code Engineering | Import Source Directory](#)

How to

To import a directory structure, using the Import Source Directory dialog, follow the steps below:

Step	Action	See also
1	<p>Select the options you require; you can configure:</p> <ul style="list-style-type: none"> • The source directory • The source type • The file extensions to look at • Whether to recurse sub directories • Whether to create a diagram for each package • Whether to import additional files as described in the Import Component Types dialog • Whether to exclude private members from libraries being imported from the model • Whether to Synchronize or Overwrite existing Classes when found; if a model Class is found matching the one in code: <ul style="list-style-type: none"> • Synchronize updates the model Class to include the details from the one in code, which preserves information not represented in code, such as the location of Classes in diagrams • Overwrite deletes the model Class and generates a new one from code, which deletes and does not replace the additional information • Whether to create a package for every directory, namespace or file; this might be restricted depending on the source type selected • How to handle Classes not found during the import (prompt for action enables you to review Classes individually) • What is shown on diagrams created by the import 	<p>Classes not found during Import^[1798]</p>
2	Click on the OK button to start.	

Learning Center topics

- **(Alt + F1) | Software Engineering | Import Code | Import Source Directory**

10.6.4 Import Binary Module

Enterprise Architect enables you to reverse-engineer certain types of binary modules.

Access [Right-click on a package within the Project Browser | Code Engineering | Import Binary Module](#)

Topics

Topic	Detail	See also
Usage	<p>Currently the permitted types are as follows:</p> <ul style="list-style-type: none"> • Java Archive (.jar) • .Net PE file (.exe, .dll) - Native Windows DLL and EXE files are not supported, only PE files containing .NET assembly data • Intermediate Language file (.il) <p>Enterprise Architect creates the necessary packages and diagrams during the import process; selecting the Do not import private members checkbox excludes private members from libraries from being imported into the model.</p> <p>When importing .Net files, you can import via reflection or via disassembly, or let Enterprise Architect decide the best method - this might result in both types being used.</p> <p>The reflection-based importer relies on a .Net program, and requires the .Net runtime environment to be installed.</p> <p>The disassembler-based importer relies on a native Windows program called <i>lldasm.exe</i>, which is a tool provided with the MS .Net SDK; the SDK can be downloaded from the Microsoft website.</p> <p>A choice of import methods is available because some files are not compatible with reflection (such as <i>mscorlib.dll</i>) and can only be opened using the disassembler; however, the reflection-based importer is generally much faster.</p> <p>You can also configure:</p> <ul style="list-style-type: none"> • Whether to Synchronize or Overwrite existing Classes when found; if a model Class is found matching the one in the file: <ul style="list-style-type: none"> • Synchronize updates the model Class to include the details from the one in the file, which preserves information not represented in the file, such as the location of Classes in diagrams • Overwrite deletes the model Class and generates a new one from the file, which deletes and does not replace the additional information • Whether to create a diagram for each package • What is shown on diagrams created by the import 	

[Learning Center topics](#)

- (Alt + F1) | [Software Engineering](#) | [Import Code](#) | [Import Compiled Java Libraries](#)
- (Alt + F1) | [Software Engineering](#) | [Import Code](#) | [Import from .Net Assemblies](#)

10.6.5 Classes Not Found During Import

Topics

Topic	Detail	See also
Usage	<p>When reverse engineering from your code, there might be times when Classes are deliberately removed from your source code.</p> <p>Enterprise Architect's <i>Import Source Directory</i> functionality keeps track of the Classes it expects to synchronize with and, on the Import Directory Structure dialog, provides options for how to handle the Classes that weren't found.</p> <p>You can select the appropriate option to make Enterprise Architect, at the end of the import, ignore the missing Classes, automatically delete them or prompt you to manage them.</p> <p>On the Import Directory Structure dialog, if you select the Prompt For Action radio button to manually review missing Classes, a dialog displays on which you specify the handling for each Class that was missing in the imported code.</p> <p>By default, all Classes are marked for deletion; to keep one or more Classes, select them and click on the Ignore button.</p>	Import a Directory Structure ^[1795]

10.7 Editing Source Code

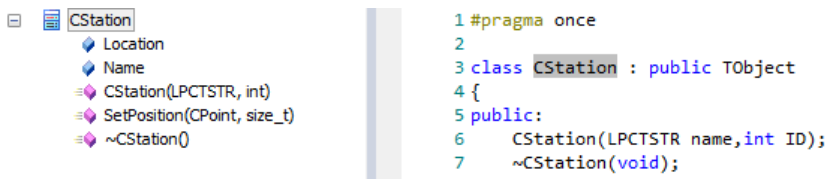
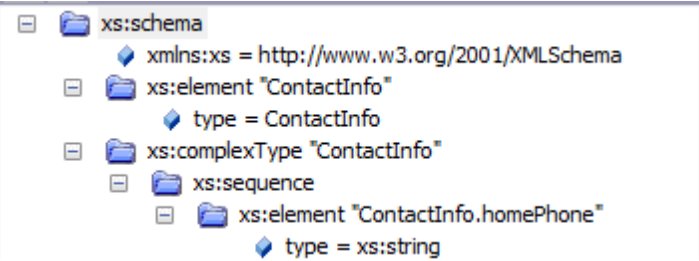
Use the Source Code editors to view and edit any source code files for an element. If a Class is selected in a diagram or the Project Browser, the Source Code editor shows the source code for that Class, provided it has already been *generated*. For C++ two tabs display, to show the implementation and header files.

A number of options change the way the Source Code editor works. They can be altered using the Options dialog (select the **Tools | Options | Source Code Engineering | Code Editors** menu option). There are also many options for developing the code, available through the Code Editor context menu, Toolbar or function keys.

There are several variations on the Source Code editor, with different access methods. The variants are discussed in the *Compare Editors* topic.

Access **Project Browser element context menu | View Source Code (F12)** or **(Ctrl+E)**
Diagram element context menu | View Source Code
View | More Element Tools | Source Code (Alt+7)
Tools | Open Source File (Ctrl+Alt+O) (files not imported to or created in Enterprise Architect)

Topics

Topic	Detail	See also
Source Code editor	<p>By default the Source Code editor is set to:</p> <ul style="list-style-type: none"> Parse all opened files, and show a tree of the results Show line numbers  <p>If you are editing an XML file, the structure tree is presented in a folder hierarchy rather than a Class structure hierarchy, as follows:</p> 	
File Parsing	<p>The Source Code editor parses files for a number of reasons; the first is to enable it to jump to the location in the file at which the currently selected item is found (but not when editing external files).</p> <p>Additionally, parsing displays a structure tree showing an overview of the file in a similar fashion to the main Project Browser; you can also select anything</p>	

Topic	Detail	See also
	in that and jump to the appropriate line in the editor.	

Notes

- There are no *context menu options* to display XML or DDL.sql files; select the appropriate object and use the keyboard keys (**F12, Alt+7**)
- When you select an element to view source code, if the element does not have a generation file (that is, code has not been or cannot be generated, such as for a Use Case), Enterprise Architect checks whether the element has a link to either an operation or an attribute of another element - if such a link exists, and that other element has source code, the code for *that* element displays
- You can also locate the directory containing a source file that has been created in or imported to Enterprise Architect, and edit it or its related files using an *external* editor such as Notepad or Visual Studio; right-click on the element in the Project Browser and select the **Open Source Directory** context menu option (**Ctrl+Alt+Y**)

Learn more

- [Generate Source Code](#)^[1768]
- [Languages Supported](#)^[1800]
- [Compare Editors](#)^[1801]
- [Code Editor Context Menu](#)^[1805]
- [Code Editor Toolbar](#)^[1802]
- [Code Editor Functions](#)^[1809]
- [Code Editor Key Bindings](#)^[1819]
- [Options - Code Editors](#)^[1895]
- [Connect to Element Feature](#)^[944]

Learning Center topics

- **(Alt + F1) | Software Engineering | Code Editor**

10.7.1 Languages Supported

The Source Code Editors can display code in all of the languages listed in the table below. For each language, the editor highlights - in colored text - the standard code syntax.

Topic	Detail	See also
Supported Languages	<ul style="list-style-type: none"> • Ada (.ada, .ads, .adb) • ActionScript (.as) 	Generate BPEL 1.1 ^[1594] Generate BPEL 2.0 ^[1618]

Topic	Detail	See also
	<ul style="list-style-type: none"> • BPEL Document (.bpel) • C++ (.h, .hh, .hpp, .c, .cpp, .cxx) • C# (.cs) • DDL Structured Query Language (.sql) • Delphi/Pascal (.pas) • Diff/Patch Files (.diff, .patch) • Document Type Definition (.dtd) • DOS Batch Files (.bat) • DOS Command Scripts (.cmd) • HTML (.html) • Interface Definition Language (.idl, .odl) • Java (.java) • Javascript (.javascript) • JScript (.js) • Modified Backus-Naur Form Grammar (.mbnf) • PHP (.php, .php4, .inc) • Python (.py) • Standard Generalized Markup Language (.sgml) • SystemC (.sc) • Visual Basic 6 (.bas) • VB.NET (.vb) • VBScript (.vbs) • Verilog (.v) • VHSIC Hardware Description Language (.vhdl) • Visual Studio Resource Configuration (.rc) • XML (eXtensible Markup Language) (.xml) 	<p>Generate DDL ^[1978]</p> <p>Web Style Templates ^[2355]</p> <p>Generate XSD ^[2049]</p>

Learn more

- [Function Details \(Syntax Highlighting\)](#) ^[1810]
- [Options - Code Editors](#) ^[1895]
- [Language Options](#) ^[1907]

10.7.2 Compare Editors

Enterprise Architect provides four principal code editor variants, available through a number of access paths. The most direct access options are identified in the descriptions below.

The first three code editor variants listed have the same display format, option toolbar, context menu options and internal function keys. They differ in their method of access and display mechanism.

Reference

Editor Variant	Function Key/ Menu Option	Description
Source Code window (Dockable)	Alt+7 View Source Code	Displays the contents of the source file for a selected Class (except if the language is C++, when the window displays a tab for the Header file and a tab for the Implementation file). If you select a different Class, the window <i>changes</i> to show the code for the new Class (unless the first Class calls the second, in which case the window scrolls down to the second Class's code instead).
Source Code View	F12 Ctrl+E Class context menu View Source Code	Displays the code on a tab of the Diagram View; the tab label shows the file name and extension (such as .java); again, for C++, there are two tabs for the Header and Implementation files. You can display the source code for other Classes on <i>additional</i> tabs, by reselecting the menu option/keys on the next Class.
Internal Editor, External Source Code	Ctrl+Alt+O Tools Open Source File	Use this option if you intend to edit external code, XML or DDL files (that is, code not imported to or generated in Enterprise Architect). Displays an external browser, then opens the specific selected code file as a tab of the Diagram View (for C++, not two code files); otherwise this is identical to the F12 option.
External Editor, Internal or External Source Code	Ctrl + Alt + Y Class context menu Open Source Directory	Displays an external file browser, open to the directory containing the selected Class's source files; you can open the files in Notepad, Visual Studio or other tools you might have on your system.

Learn more

- [Code Editor Context Menu](#)^[1805]
- [Code Editor Toolbar](#)^[1802]
- [Code Editor Functions](#)^[1809]
- [Code Editor Key Bindings](#)^[1819]
- [Options - Code Editors](#)^[1895]

10.7.3 Code Editor Toolbar



The toolbar buttons in the Source Code editor enable you to change, view and interact with the code contained in the editor. The function of each button, from left to right, is described below:

The Toolbar:



Topics

Topic	Detail	See also
Toolbar Options	<ul style="list-style-type: none"> • Structure Tree - shows or hides the element hierarchy panel (the left panel of the Source Code editor) • Line Numbers - shows or hides the line numbers against the lines of code • Source Code Engineering Properties - displays options to select Source Code Engineering pages of the Options dialog, from which you can configure display and behavior options for source code engineering: <ul style="list-style-type: none"> • Syntax Highlighting options • DDL and Code Editor options • Code Engineering options • Code Editor Key Bindings • Editor Functions - provides quick access to the following functions: <ul style="list-style-type: none"> • Open Corresponding File - opens the header or implementation file associated with the currently-open file • Go to Matching Brace - for a selected opening or closing brace, highlights the corresponding closing or opening brace in the pair • Go to Line - displays a small dialog on which you select the number of the line to highlight; click on the OK button to move the cursor to that line • Cursor History Previous - the Source Code viewer keeps a history of the previous 50 cursor positions, creating a record when the cursor is moved either more than 10 lines away from its previous position, or in a find-and-replace operation; the menu option moves the cursor to the position in the immediately-previous cursor history record • Cursor History Next - moves the cursor to the position in the immediately-following cursor history record • Record Macro - record the subsequent keystrokes to be saved as a macro • Stop Recording and Save Macro - stop recording the keystrokes and specify a name for the macro, on the 	<p>Source Code Engineering Options <small>[1891]</small></p> <p>Editor Language Properties <small>[1896]</small></p> <p>Options - Code Editors <small>[1895]</small></p> <p>Source Code Options <small>[1892]</small></p> <p>Code Editor Key Bindings <small>[1819]</small></p> <p>Code Templates Editor <small>[1404]</small></p>

Topic	Detail	See also
	<p>Save Macro dialog</p> <ul style="list-style-type: none"> • Play Macro - execute the macro to repeat the saved keystrokes, if necessary selecting the macro from the Open Macro dialog • Toggle Line Comment - comments out (//) or re-establishes the code for each full line in which text is highlighted • Toggle Stream Comment - inserts a stream comment (/ * */) at the cursor position or comments out the highlighted characters and lines, or re-establishes the commented text as code • Toggle Whitespace Characters - shows or hides the spacing characters: --> (tab space) and . (character space) • Toggle EOL Characters - shows or hides the end-of-line characters: CR (carriage return) and LF (line feed) • Save Source and Resynchronize Class - saves the source code and resynchronizes the Class • Code Templates - accesses the Code Templates Editor • Find in Project Browser - for a selected line of code, highlights the appropriate structure in the Project Browser; if there is more than one possibility the Possible Matches dialog displays, listing the occurrences of the appropriate structure from which you can select the required one • Search in Files - searches for the selected object name in associated files and displays the results of the search on the File Search window • Search in Model - searches for the selected text throughout the model, and displays the results of the search on the Model Search window • Go to Declaration - locates the declaration of a symbol in the source code • Go to Definition - locates the definition of a symbol in the source code (applicable to languages where symbols are declared and defined in separate files, such as C++, Delphi) • Autocomplete List - displays the autocomplete list of possible values; double-click on a value to select it 	<p>Find in Files  ¹⁸¹⁷</p> <p>Model Search  ⁶¹⁷</p>

Topic	Detail	See also
	<ul style="list-style-type: none"> • Parameter Information - when the cursor is between the parentheses of an operation's parameter list, displays the operation's signature, highlighting the current parameter • Find Current Class in Project Browser - displays the name of the currently-selected Class in the code, and highlights that name in the Project Browser; if there is more than one possibility the Possible Matches dialog displays, listing the occurrences of the Class from which you can select the required one • Find Member - displays the name of the currently-selected attribute or method in the code, and highlights that name in the Project Browser; if there is more than one possibility the Possible Matches dialog displays, listing the occurrences of the feature from which you can select the required one 	

Notes

- The **Record Macro** function disables Intellisense while the macro is being recorded
- You can **assign key strokes** to execute the macro, instead of using the toolbar drop-down and Open Macro dialog

Learn more

- [Intellisense](#)^[1812]
- [Editor Language Properties](#)^[1896]


10.7.4 Code Editor Context Menu

When working on a file with a code editor, you can access a number of options through the context menu. Right-click on a text string to display the menu.

Topics

Topic	Detail	See also
Options	<p>The options on the menu can vary depending on which editor you are using, but should include some or all of the following:</p> <ul style="list-style-type: none"> • Go to Declaration locates the declaration of a symbol in the source code • Go to Definition locates the definition of a symbol in the source 	

Topic	Detail	See also
	<p>code (applicable to languages where symbols are declared and defined in separate places, such as C++ or Delphi)</p> <ul style="list-style-type: none"> • Search for '<string>' - displays a submenu that enables you to locate the search string in a range of locations: <ul style="list-style-type: none"> • Find in Project Browser finds the object containing the selected text in the Project Browser • Search in Open Files opens the Execution Analyzer File Search facility, then searches for the selected text string in other code files that are open of the same type and in the same folder as the current file, displaying the results in Tree View; you can change the folder path, search text and file type as required within the File Search window • Search in Files performs the same search as Search in Open Files, except that the search is in all comparable files whether they are open or not • Search in Model performs an <i>Element Name</i> search in the Model Search facility, and displays the results on the Model Search tab • EA User Guide displays the description of the code item in the <i>Enterprise Architect User Guide</i> • Google displays the results of a search on the text from a Google search • MSDN displays the results of a search on the text in the Microsoft Developer Network (MSDN) • Sun Java SE displays the results of a search on the text in the Sun Microsystems 'Sun Search' facility • Wikipedia displays any entry on the object on the <i>Wikipedia</i> web site • Koders displays the results of the search for the text string on <i>Koders.com</i> • Set Debugger to Line - (if the debugger is executing and has reached a breakpoint) moves the execution point to the current line; ensure that you do not skip over any code or declarations that affect the next section of code being debugged • Display Variable - (if the debugger is executing) opens the Locals window and highlights the local variable for the current point in the code • Create Use Case for '<string>' - displays the Create Use Case 	<p>Find In Files ^[1817]</p> <p>Model Search ^[617]</p> <p>Pre-Defined Searches ^[617]</p> <p>View the Local Variables ^[1877]</p> <p>Create Use Case For Method ^[1807]</p> <p>Set Record Markers ^[2155]</p>

Topic	Detail	See also
	<p>For Method dialog, through which you create a Use Case for the method containing the text string</p> <ul style="list-style-type: none"> • Breakpoint - displays a submenu of options for creating a breakpoint or recording marker on the selected line of code • Open (Close) IME - opens the Input Method Editor to enable you to enter text in your selected foreign language script (such as Japanese); you set the keyboard language using the Windows Control Panel - Regional and Language Options facility • Line Numbers - (Script Editor only) shows or hides the code line numbers on the left hand side of the editor screen <p>The remaining six context menu options provide simple editing functions.</p>	
Script Editor	<p>If you select the Search for '<string>' context menu option while working in the Script Editor, a slightly different submenu displays.</p> <p>The Search in Scripts option opens the Execution Analyzer File Search facility, setting the Search Path field to Search in Scripts and the Search Text field to the selected text, then searching all scripts for the text string and displaying the results of the search in Tree View.</p>	Script Editor 

Notes

- The options in the lower half of the menu (after **Search in Model**) are configurable; you can add new search tools or remove existing ones by editing the *searchProviders.xml* file in the *Sparx Systems > EA > Config* folder - this file is in OpenSearch description document format

Learn more

- [OpenSearch Description Document Format](#) (Online Resource)

10.7.4.1 Create Use Case for Method

The code editor context menu enables you to create a Use Case element for a method that you select from the code.

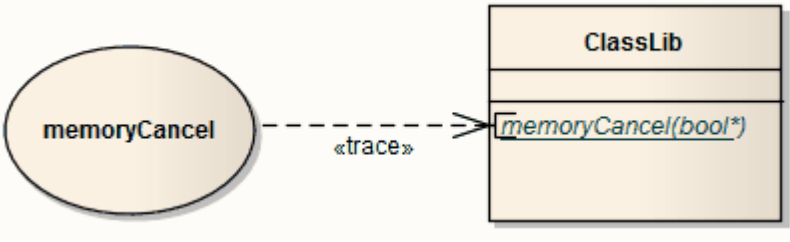
You can also:

- Link the Use Case directly to the method
- Add the parent Class to a diagram (if it is not already in the selected diagram) and/or add the Use Case element to the diagram
- Block from display any attributes or methods that are not also the targets of feature links

How to

To create a Use Case for a method, through the code editor

Step	Action	See also
1	(If you want to depict the Use Case and its link to the method <i>in a diagram</i>) click on the diagram name in the Project Browser.	
2	In the code editor, right click on either the method name or any part of the method body, and select the Create Method for <methodname> context menu option. The Create Use Case for Method dialog displays.	Code Editor Context Menu [1805]
3	The basic function of this dialog is to create a Use Case for the selected method: <ul style="list-style-type: none"> • If this is all that is required, click on the OK button; the Use Case element is created in the Project Browser, in the same package as the parent Class for the method, and with the same name as the method • If you intend to make the relationship tangible, continue with the procedure 	
4	To create a Trace connector linking the Use Case to the method, select the Link Use Case to Method check box.	
5	To add the method's parent Class to the diagram, if it is not already there, select the Add Class to Diagram check box.	
6	To add the newly-created Use Case to the diagram, select the Add Use Case to Diagram check box; this would now show the Use Case, Class and Trace connector on the diagram.	
7	To only show the features (attributes and methods) of the parent Class that are the targets of 'link to feature' relationships, select the Display only linked features in Class checkbox. The Class might contain any number of attributes and methods, but those without a 'link to feature' relationship are hidden.	Connect to Element Feature [944]
8	Click on the OK button to create and depict the Use Case and relationship; if you selected all options, the diagram now contains linked elements resembling the following:	

Step	Action	See also
		

10.7.5 Code Editor Functions

Topic	Detail	See also
Code Editor	<p>The common Code Editor provides a variety of functions to assist with the code editing process, including:</p> <ul style="list-style-type: none"> • Syntax Highlighting • Bookmarks • Cursor History • Brace Matching • Automatic Indentation • Commenting Selections • Scope Guides • Zooming • Line Selection • intellisense • Find and Replace • Find in Files <p>A range of these functions is available through keyboard key combinations and/or context menu options.</p> <p>You can customize several of the Code Editor features by setting properties in the Code Editor configuration files; for example, by default the line containing the cursor is always highlighted, but you can turn the highlighting off.</p>	<p>Function Details ^[1810]</p> <p>intellisense ^[1812]</p> <p>Find and Replace ^[1814]</p> <p>Find in Files ^[1817]</p> <p>Code Editor Key Bindings ^[1819]</p> <p>Code Editor Context Menu ^[1805]</p>

Learning Center topics

- (Alt + F1) | **Software Engineering | Code Editor | Locate in Source**
- (Alt + F1) | **Software Engineering | Code Editor | Using Intellisense**

10.7.5.1 Function Details

Topic	Detail	See also
Syntax Highlighting	<p>The Code Editor highlights - in colored text - the standard code syntax of all language file formats supported by Enterprise Architect</p> <pre data-bbox="421 454 898 831"> 1 #pragma once 2 #include "afxwin.h" 3 #include "afxcmn.h" 4 5 6 // CToolBox dialog 7 8 class CToolBox : public CDialog 9 { 10 DECLARE_DYNAMIC(CToolBox) 11 CRect m_rect; 12 int m_offset; </pre> <p>You can define how the Code Editor implements syntax highlighting for each language, through the Code Editors page of the Options dialog.</p>	Languages Supported ^[1800] Options - Code Editors ^[1895] Editor Language Properties ^[1896] Language Options ^[1907]
Bookmarks	<p>Bookmarks denote a line of interest in the document; you can toggle them on and off for a particular line by pressing (Ctrl+F2).</p> <p>Additionally, you can press (F2) and (Shift+F2) to navigate to the next or previous bookmark in the document.</p>	
Cursor History	<p>The Code Editor Control keeps a history of the previous 50 cursor positions; an entry in the history list is created when:</p> <ul data-bbox="469 1249 1225 1330" style="list-style-type: none"> • The cursor is moved more than 10 lines from its previous position • The cursor is moved in a <i>find/replace</i> operation <p>You can navigate to an earlier point in the cursor history by pressing (Ctrl+-), and to a later point by pressing (Ctrl+Shift+-).</p>	
Brace Matching	<p>When you place the cursor over a brace or bracket, the Code Editor highlights its corresponding partner; you can then navigate to the matching brace by pressing (Ctrl+E).</p> <pre data-bbox="437 1615 1217 1671"> 28 function ProtectedFunctionTest: boolean; 29 procedure ProtectedProcedureTest(a: WideString); </pre>	
Automatic Indentation	<p>For each supported language, the Code Editor adjusts the indentation of a new line according to the presence of control statements or scope block tokens in the lines leading up to the cursor position.</p>	

Topic	Detail	See also
	<pre data-bbox="421 320 1134 566"> 358 { 359 for(size_t t = 0; t < Stations.size(); t++) 360 { 361 if(Stations[t]->Location == loc) 362 return Stations[t]; 363 } 364 return NULL; 365 }</pre> <p data-bbox="421 589 1043 613">The levels of indent are indicated by pale horizontal lines.</p> <p data-bbox="421 636 1251 689">You can also manually indent selected lines and blocks of code by pressing (Tab); to unindent the selected code, press (Shift+Tab).</p>	
Commenting Selections	<p data-bbox="421 768 1251 822">For languages that support comments, the Code Editor can comment entire selections of code.</p> <p data-bbox="421 844 1011 869">The Code Editor recognizes two types of commenting:</p> <ul data-bbox="469 891 1219 1137" style="list-style-type: none"> • Line Commenting - entire lines are commented from the start (for example: <pre data-bbox="501 969 836 994">// This is a comment)</pre> • Stream Commenting - sections of a line are commented from a specified start point to a specified end point (for example: <pre data-bbox="501 1104 884 1128">/* This is a comment */)</pre> <p data-bbox="421 1205 1187 1229">You can toggle comments on the current line or selection by pressing:</p> <ul data-bbox="469 1252 900 1323" style="list-style-type: none"> • (Ctrl+Shift+C) for line comments, or • (Ctrl+Shift+X) for stream comments 	
Scope Guides	<p data-bbox="421 1406 1251 1487">If the cursor is placed over an indentation marker, the Code Editor performs a 'look back' to find the line that started the scope at that indentation level; if the line is found and is currently on screen, it is highlighted in light blue.</p> <pre data-bbox="421 1541 1251 1720"> -- 93 // If there were any answers, then return a packet, if not then just return nul 94 // to indicate the server has no response 95 if (answers.size() > 0) 96 { 97 DNSPacket responsePacket = Helpers.createResponsePacket(answers, this.theS 98 responsePacket.queryID = receivedPacket.queryID; 99 100 return responsePacket; 101 }</pre> <p data-bbox="421 1771 1251 1825">Alternatively if the line is off screen, a calltip is displayed advising of the line number and contents:</p>	

Topic	Detail	See also
	<pre> 93 // If there were any answers, then return a packet, if not then just return null 94 // to indicate the server has no response 95 if (answers.size() > 0) 96 Line 73: private DNSPacket processQuery(DNSPacket receivedPacket) 97 DNSPacket responsePacket = Helpers.createResponsePacket(answers, this.theSt 98 responsePacket.queryID = receivedPacket.queryID; 99 100 return responsePacket; 101 } </pre>	
Zooming	<p>You can zoom into and out of the contents of the Code Editor using:</p> <ul style="list-style-type: none"> • (Ctrl) + keypad (+) and • (Ctrl) + keypad (-) <p>Zoom can be restored to 100% using (Ctrl) + keypad (/).</p>	
Line Selection	<p>If you want to move the cursor to a specific line of code, press (Ctrl + G) and, in response to the prompt, type in the line number.</p> <p>Press the OK button; the editor displays the specified line of code with the cursor at the left.</p>	

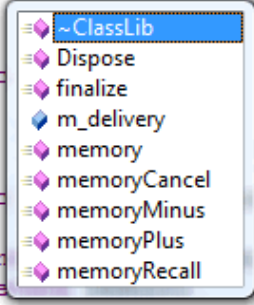
10.7.5.2 Intellisense

Intellisense is a feature that provides choices of code items and values as you type. Not all code editors use intellisense; for example, Intellisense is disabled while you record a macro in the Source Code Viewer.

Intellisense provides you with context-based assistance through autocompletion lists, calltips and mouseover information.

Topics

Topic	Detail	See also
Autocompleti on List	<p>An autocompletion list provides a list of possible completions for the current text; the list is automatically invoked when you enter an accessor token (such as a period or pointer accessor) after an object or type that contains members.</p>	

Topic	Detail	See also
	<pre data-bbox="451 320 1193 757"> 57 public void memoryRecall() 58 { 59 this. 60 } 61 62 public 63 64 } 65 66 public 67 { 68 in 69 re 70 } 71 </pre>  <p data-bbox="432 801 1246 965">You can also invoke the autocomplete list manually by pressing (Ctrl + Spacebar); the Code Editor then searches for matches for the word leading up to the invocation point.</p> <p data-bbox="432 909 1246 965">Select an item from the list and press (Enter) or (Tab) to insert the item into the code; to dismiss the autocomplete list, press (Escape).</p>	
Calltips	<p data-bbox="432 1037 1262 1155">Calltips display the current method's signature when you type the parameter list token (for example, opening parenthesis); if the method is overloaded, the calltip displays arrows that you can use to navigate through the different method signatures</p> <pre data-bbox="483 1182 1193 1529"> 20 //PostDraw Adornments 21 //Stereotyped Static Adornments 22 //Add Stakeholder's STAKE 23 setpenwidth(24 // Add a the SetPenWidth(int penwidth) 25 startpath() ; 26 moveto (25,37) ; 27 lineto (25,52) ; 28 endpath() ; 29 strokepath() ; 30 //Add tip </pre>	
Mouseover Information	<p data-bbox="432 1608 1262 1666">You can display supporting documentation for code elements (for example, attributes and methods) by hovering the cursor over the element in question.</p> <pre data-bbox="451 1720 1042 1877"> 11 dockable = "none"; 12 13 string 14 Dock elements together. Tagged V 15 Valid Values: none, standard 16 //PreDraw Derived Attribute I </pre>	

[Learn more](#)

- [Code Editor Functions](#)^[1809]
- [Editing Source Code](#)^[1799]

10.7.5.3 Find and Replace

Each of Enterprise Architect's code editors facilitates searching for and replacing terms in the editor, through the Find and Replace dialog.

Access

Highlight the required text string and press:

- **(Ctrl+F)** for the *find* controls only, or
- **(Ctrl+R)** for both *find* and *replace* controls

In each instance, the **Find what** field is populated with the text currently selected in the editor. If no text is selected in the editor, the **Find what** field is populated with the word at the current cursor position. If no word exists at the current cursor position, the last searched-for term is used.

Basic Operations:

Commands:

Command	Action	See also
Find Next	Locate and highlight the next instance (relative to the current cursor position) of the text specified in the Find what field.	
Replace	Replace the current instance of the text specified in the Find what field with the text specified in the Replace with field, and then locate and highlight the next instance (relative to the current cursor position) of the text specified in the Find what field.	
Replace All	Automatically replace all instances of the text specified in the Find what field with the text specified in the Replace with field.	

Options:

Option	Action	See also
Match Case	Specify that the case of each character in the text string in the Find what field is significant when searching for matches in the code.	
Match whole word	Specify that the text string in the Find what field is a complete word and should not be matched with instances of the text that form part of a longer string. For example, searches for ARE should not match those letters in instances of the words AREA or ARENA.	

Option	Action	See also
Search up	Perform the search from the current cursor position up to the start of the file, rather than in the default direction of current cursor position to end of file.	
Use Regular Expressions	Evaluate specific character sequences in the Find what and Replace with fields as <i>Regular Expressions</i> (see below).	

Topics

Topic	Detail	See also										
Regular Expressions	<p>A Regular Expression is a formal definition of a search pattern, which can be used to match specific characters, words or patterns of characters.</p> <p>For the sake of simplicity, the Code Editor's 'find and replace' mechanism supports only a subset of the standard Regular Expression grammar.</p> <p>Text in the Find what and Replace with fields is only interpreted as a Regular Expression if the Use Regular Expressions checkbox is selected in the Find and Replace dialog.</p>											
Meta sequences	<p>If the Use Regular Expressions checkbox is selected, most characters in the Find what field are treated as literals (that is, they match only themselves).</p> <p>The exceptions are called <i>metasequences</i>; each <i>metasequence</i> recognized in the Code Editor Find and Replace dialog is described in the following table:</p> <table border="1"> <thead> <tr> <th>Metasequence</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>.</td> <td>Represents any single character. For example: .at is matched to cat, hat, bat and rat.</td> </tr> <tr> <td>\<</td> <td>Indicates that the text is the start of a word. For example: \<cat is matched to catastrophe and cataclysm, but not concatenate.</td> </tr> <tr> <td>\></td> <td>Indicates that the text is the end of a word. For example: hat> is matched to that and chat, but not hate.</td> </tr> <tr> <td>(...)</td> <td>Indicates alternative single characters that can be</td> </tr> </tbody> </table>	Metasequence	Description	.	Represents any single character. For example: .at is matched to cat , hat , bat and rat .	\<	Indicates that the text is the start of a word. For example: \<cat is matched to catastrophe and cataclysm , but not concatenate .	\>	Indicates that the text is the end of a word. For example: hat> is matched to that and chat , but not hate .	(...)	Indicates alternative single characters that can be	
Metasequence	Description											
.	Represents any single character. For example: .at is matched to cat , hat , bat and rat .											
\<	Indicates that the text is the start of a word. For example: \<cat is matched to catastrophe and cataclysm , but not concatenate .											
\>	Indicates that the text is the end of a word. For example: hat> is matched to that and chat , but not hate .											
(...)	Indicates alternative single characters that can be											

Topic	Detail	See also														
	<table border="1"> <thead> <tr> <th data-bbox="520 309 635 398">Metasequence</th> <th data-bbox="635 309 1286 398">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="520 398 635 678"></td> <td data-bbox="635 398 1286 678"> <p>matched. The characters can be specific - (chr) - or in an alphabetical or numerical range - (a-m).</p> <p>For example: (hc) at is matched to hat and cat but not bat.</p> <p>For example: (a-m) Class is matched to any name in the range aClass - mClass.</p> </td> </tr> <tr> <td data-bbox="520 678 635 1048">(^...)</td> <td data-bbox="635 678 1286 1048"> <p>Indicates alternative single characters that should be excluded from a match. The characters can be specific - (^chr) - or in an alphabetical or numerical range - (^a-m).</p> <p>For example: (^hc) at is matched to rat and bat but hat and cat are excluded.</p> <p>For example: (^a-m) Class is matched to any name in the range nClass to zClass but aClass to mClass are excluded.</p> </td> </tr> <tr> <td data-bbox="520 1048 635 1149">^</td> <td data-bbox="635 1048 1286 1149">Matches the start of a line.</td> </tr> <tr> <td data-bbox="520 1149 635 1249">\$</td> <td data-bbox="635 1149 1286 1249">Matches the end of a line.</td> </tr> <tr> <td data-bbox="520 1249 635 1529">*</td> <td data-bbox="635 1249 1286 1529"> <p>Matches the preceding character (or character set) 0 or more times.</p> <p>For example: ba*t is matched to bt, bat, baat, baaat and so on.</p> <p>For example: b(ea) *t is matched to bt, bet, bat, beat, beet, baat and so on.</p> </td> </tr> <tr> <td data-bbox="520 1529 635 1798">+</td> <td data-bbox="635 1529 1286 1798"> <p>Matches the preceding character (or character set) 1 or more times.</p> <p>For example: ba+t is matched to bat, baat and baaat but not bt.</p> <p>For example: b(ea) +t is matched to bet, bat, beat, beet and baat but not bt.</p> </td> </tr> </tbody> </table> <p data-bbox="520 1816 1286 1906">If a single character metasequence is preceded by a backslash (\) it is treated as a literal character (at\) matches c(at) as the brackets are treated literally.</p> <p data-bbox="520 1921 1286 2042">When the Use Regular Expressions checkbox is selected, a metasequence helper menu is available to the right of both of the Find what and Replace with fields; selecting a metasequence from this menu inserts the metasequence into the field, replacing or wrapping</p>	Metasequence	Description		<p>matched. The characters can be specific - (chr) - or in an alphabetical or numerical range - (a-m).</p> <p>For example: (hc) at is matched to hat and cat but not bat.</p> <p>For example: (a-m) Class is matched to any name in the range aClass - mClass.</p>	(^...)	<p>Indicates alternative single characters that should be excluded from a match. The characters can be specific - (^chr) - or in an alphabetical or numerical range - (^a-m).</p> <p>For example: (^hc) at is matched to rat and bat but hat and cat are excluded.</p> <p>For example: (^a-m) Class is matched to any name in the range nClass to zClass but aClass to mClass are excluded.</p>	^	Matches the start of a line.	\$	Matches the end of a line.	*	<p>Matches the preceding character (or character set) 0 or more times.</p> <p>For example: ba*t is matched to bt, bat, baat, baaat and so on.</p> <p>For example: b(ea) *t is matched to bt, bet, bat, beat, beet, baat and so on.</p>	+	<p>Matches the preceding character (or character set) 1 or more times.</p> <p>For example: ba+t is matched to bat, baat and baaat but not bt.</p> <p>For example: b(ea) +t is matched to bet, bat, beat, beet and baat but not bt.</p>	
Metasequence	Description															
	<p>matched. The characters can be specific - (chr) - or in an alphabetical or numerical range - (a-m).</p> <p>For example: (hc) at is matched to hat and cat but not bat.</p> <p>For example: (a-m) Class is matched to any name in the range aClass - mClass.</p>															
(^...)	<p>Indicates alternative single characters that should be excluded from a match. The characters can be specific - (^chr) - or in an alphabetical or numerical range - (^a-m).</p> <p>For example: (^hc) at is matched to rat and bat but hat and cat are excluded.</p> <p>For example: (^a-m) Class is matched to any name in the range nClass to zClass but aClass to mClass are excluded.</p>															
^	Matches the start of a line.															
\$	Matches the end of a line.															
*	<p>Matches the preceding character (or character set) 0 or more times.</p> <p>For example: ba*t is matched to bt, bat, baat, baaat and so on.</p> <p>For example: b(ea) *t is matched to bt, bet, bat, beat, beet, baat and so on.</p>															
+	<p>Matches the preceding character (or character set) 1 or more times.</p> <p>For example: ba+t is matched to bat, baat and baaat but not bt.</p> <p>For example: b(ea) +t is matched to bet, bat, beat, beet and baat but not bt.</p>															

Topic	Detail	See also
	the currently selected text as appropriate.	
Tagged Regions	<p>When 'find and replacing' with Regular Expressions, up to nine sections of the original term can be substituted into the replacement term.</p> <p>The metasequences <code>\(</code> and <code>\)</code> denote the start and the end of a <i>tagged region</i>; the section of the matched text that falls within the tagged region can be included in the replacement text with the metasequence <code>\n</code> (where <i>n</i> is the tagged region number between 1 and 9).</p> <p>For example:</p> <p>Find what: <code>\((A-Za-z) +\) ' s t h i n g s</code></p> <p>Replace with <code>i t e m s t h a t b e l o n g t o \ 1</code></p> <p>Original text: These are all Michael's things.</p> <p>Replaced text: These are all items that belong to Michael</p>	

10.7.5.4 Find in Files

This topic describes how to use the File Search facility to search files for data names and structures.

Access [Edit | File Search](#) (Ctrl+Shift+Alt+F)

Topics

Topic	Detail	See also
Usage	<p>File Text Searches are provided by the Search window and from within the Code Editors.</p> <p>The Search window enables you to search for text in code files and scripts; you can select to display the results of the search in one of two formats:</p> <ul style="list-style-type: none"> • List View - each result line consists of the file path and line number, followed by the line text; multiple lines from one file are listed as separate entries • Tree View - each result line consists of the file path that matches the search criteria, and the number of lines matching the search text within that file; you can expand the entry to show the line number and text of each line 	

Search Toolbar:

You can use the toolbar options in the Search window to control the search operation. The state of all buttons persists over time to always reflect your previous search criteria.



The options, from left to right, are described below.

Reference

Field/Option	Usage	See also
Search Path list box	<p>Specify the folder to search.</p> <p>You can type the path to search directly into the text box, or click on the folder icon to browse for the path.</p> <p>Any paths you enter are automatically saved in the drop-down list, up to a maximum of ten; paths added after that overwrite the oldest path in the list.</p> <p>A fixed option in the drop-down list is Search in Scripts, which sets the search to operate on all local and user-defined scripts in the Scripts tab of the Scripting window; this option disables the Search File Types list box.</p>	
Search Text list box	<p>Specify the text to look for.</p> <p>You can type the text directly into the text box or click on the drop-down arrow to select from a previous entry in the list. The search text you enter is automatically saved in the list when you click on the Search button.</p> <p>The list box saves up to ten search queries. Search queries added after that overwrite the oldest query in the list.</p>	
Search File Types list box	<p>Limit the search to specific types of files. You can select multiple file types in a string, separated by either a comma or a semi-colon as shown in the image above.</p>	
Search button	<p>Begin the search.</p> <p>During the course of the search all other buttons in the toolbar are disabled. You can cancel the search at any time by clicking on the Search button again.</p> <p>If you switch any of the toggle buttons below, you must run the search again to change the output.</p>	
Case Sensitivity button	<p>Toggle the case sensitivity of the search. The tooltip message identifies the current status of the button.</p>	
Word Match button	<p>Toggle between searching for any match and searching for only those matches that form an entire word. The tooltip message</p>	

Field/Option	Usage	See also
	identifies the current status of the button.	
SubFolders button	Toggle between limiting the search to a single path and including all subfolders under that path. The tooltip message identifies the current status of the button.	
Result View button	Select the presentation format of the search results - List View or Tree View format.	
Clear Results button	Clear the results.	
Clear Search Criteria button	Remove all the entries in the Search Path , Search Text and Search File Types list boxes, if required.	
Help button	Display this Help topic.	

Learning Center topics

- (Alt + F1) | [Software Engineering](#) | [Code Editor](#) | [Search for Options](#)

10.7.6 Code Editor Key Bindings

Key	Description
Ctrl + G	Move cursor to a specified line
↓	Move cursor down one line
Shift + ↓	Extend selection down one line
Ctrl + ↓	Scroll down one line
Alt + Shift + ↓	Extend rectangular selection down one line
↑	Move cursor up one line
Shift + ↑	Extend selection up one line
Ctrl + ↑	Scroll up one line
Alt + Shift + ↑	Extend rectangular selection up one line
Ctrl+ (Move cursor up one paragraph

Key	Description
Ctrl+Shift+ (Extend selection up one paragraph
Ctrl+)	Move cursor down one paragraph
Ctrl+Shift+)	Extend selection down one paragraph
←	Move cursor left one character
Shift + ←	Extend selection left one character
Ctrl + ←	Move cursor left one word
Ctrl + Shift + ←	Extend selection left one word
Alt + Shift + ←	Extend rectangular selection left one character
→	Move cursor right one character.
Shift + →	Extend selection right one character
Ctrl + →	Move cursor right one word
Ctrl + Shift + →	Extend selection right one word
Alt + Shift + →	Extend rectangular selection right one character
Ctrl + /	Move cursor left one word part
Ctrl + Shift + /	Extend selection left one word part
Ctrl + \	Move cursor right one word part
Ctrl + Shift + \	Extend selection right one word part
Home	Move cursor to the start of the current line
Shift + Home	Extend selection to the start of the current line
Ctrl + Home	Move cursor to the start of the document
Ctrl + Shift + Home	Extend selection to the start of the document
Alt + Home	Move cursor to the absolute start of the line
Alt + Shift + Home	Extend rectangular selection to the start of the line
End	Move cursor to the end of the current line
Shift + End	Extend selection to the end of the current line
Ctrl + End	Move cursor to the end of the document
Ctrl + Shift + End	Extend selection to the end of the document

Key	Description
Alt + End	Move cursor to the absolute end of the line
Alt + Shift + End	Extend rectangular selection to the end of the line
Page Up	Move cursor up a page
Shift + Page Up	Extend selection up a page
Alt + Shift + Page Up	Extend rectangular selection up a page
Page Down	Move cursor down a page
Shift + Page Down	Extend selection down a page
Alt + Shift + Page Down	Extend rectangular selection down a page
Delete	Delete character to the right of the cursor
Shift + Delete	Cut selection
Ctrl + Delete	Delete word to the right of the cursor
Ctrl + Shift + Delete	Delete until the end of the line
Insert	Toggle overtype
Shift + Insert	Paste
Ctrl + Insert	Copy selection
Backspace	Delete character to the left of the cursor
Shift + Backspace	Delete character to the left of the cursor
Ctrl + Backspace	Delete word to the left of the cursor
Ctrl + Shift + Backspace	Delete from the start of the line to the cursor
Alt + Backspace	Undo delete
Tab	Indent cursor one tab
Ctrl + Shift + I	Indent cursor one tab
Shift + Tab	Unindent cursor one tab
Ctrl + keypad(+)	Zoom in
Ctrl + keypad(-)	Zoom out

Key	Description
Ctrl + keypad(/)	Restore Zoom
Ctrl + Z	Undo
Ctrl + Y	Redo
Ctrl + X	Cut selection
Ctrl + C	Copy selection
Ctrl + V	Paste
Ctrl + L	Cut line
Ctrl + Shift + L	Delete line
Ctrl + T	Transpose line
Ctrl + Shift + T	Copy line
Ctrl + A	Select entire document
Ctrl + D	Duplicate selection
Ctrl + U	Convert selection to lowercase
Ctrl + Shift + U	Convert selection to uppercase
Ctrl + E	Move cursor to matching brace
Ctrl + Shift + E	Extend selection to matching brace
Ctrl + Shift + C	Toggle line comment on selection
Ctrl + Shift + X	Toggle stream comment on selection
Ctrl + F2	Toggle bookmark
F2	Go to next bookmark
Shift + F2	Go to previous bookmark
Ctrl + Shift + W	Toggle whitespace characters
Ctrl + Shift + L	Toggle EOL characters
Ctrl + Spacebar	Invoke autocomplete

Key	Description
Ctrl + -	Go backwards in cursor history
Ctrl + Shift + -	Go forwards in cursor history
F12	Start/Cancel search for keyword in file(s).
Ctrl + F	Find text
Ctrl + R	Replace text

Notes

- In addition to the following keys, you can assign (**Ctrl + Alt + n**) key combinations to macros that you define within the Source Code Editor

Learn more

- [Editor Language Properties](#)^[1896]
- [Source Code Viewer Toolbar](#)^[1802]

10.8 Code Build & Debug



This section discusses the integrated Model Driven Development Environment (MDDE).

Topics

Topic	Detail	See also
Abstract	<p>Model Driven Development provides a more robust, accessible and faster development cycle than traditional coding-driven cycles.</p> <p>A well constructed model, intimately linked with source code build, run, debug, test and deploy capabilities provides a rich, easily navigated and easily understood target architecture.</p> <p>Traceability, linkage to Use Cases, Components and other model artifacts, plus the ability to readily record and document pre-existing or recently developed code, make Enterprise Architect's development environment uniquely powerful.</p> <p>Enterprise Architect incorporates industry standard intelligent editing, debuggers and modeling languages.</p>	<p>Getting Started ^[1825]</p> <p>Analyzer Scripts ^[1827]</p> <p>Editing Source Code ^[1799]</p> <p>Run a Debug Session ^[1890]</p>
The MDDE	<p>The MDDE provides tools to design, build and debug an application:</p> <ul style="list-style-type: none"> • UML technologies and tools to model software • Code generation tools to generate/reverse engineer source code • Tools to import source code and binaries • Code editors that support different programming languages • Intellisense to aid coding • Analyzer scripts that enable a user to describe how to build, debug, test and deploy the application <pre>pApp = new CBCGPAAppointmentDemo { ▲ 2 of 2 ▼ CBCGPAAppointmentDemo::CBCGPAAppointmentDemo(COLEDateTime& dtStart, COLEDateTime& dtFinish, CString& strText, COLORREF clrBackground, COLORREF clrForeground, COLORREF clrDuration) RGB (165, 222, 99), CLR_DEFAULT, RGB(128, 0, 128) };</pre>	<p>Development Tools ^[1734]</p> <p>MDG Technologies ^[1283]</p> <p>Modeling Basics ^[657]</p> <p>Generate Source Code ^[1768]</p> <p>Import Source Code ^[1792]</p> <p>Import Binary Module ^[1797]</p> <p>Editing Source Code ^[1799]</p> <p>Intellisense ^[1812]</p> <p>Managing Analyzer Scripts ^[1828]</p>

10.8.1 Getting Started

To quickly start development in the Model Driven Development Environment, check through the following topics:

- [Prerequisites](#)^[1825]
- [General Workflow](#)^[1825]
- [Workspace Layout and Tools](#)^[1826]

10.8.1.1 Prerequisites

Before using the Model Driven Development Environment:

- You should be using the correct edition: Enterprise Architect Professional, Corporate or extended editions
- You should be connected to the required model
- Relevant source code files should be linked to Classes in the model (sufficient to be editable from the internal code editors) either:
 - Using MDA Transforms to generate stub code files, and internal editing to write code
 - Reverse engineering source code from existing files, or
 - Importing existing source code from external files
- Required External Frameworks or compilers should be installed and operable
- An Analyzer Script should be configured to link to the compilers in the External Framework

Learn more

- [Analyzer Scripts](#)^[1827]
- [Importing Source Code](#)^[1790]
- [Editing Source Code](#)^[1799]
- [Add Build Commands](#)^[1832]
- [Debugging](#)^[1867]

Learning Center topics

- (Alt+F1) | **Software Engineering | Import Code**
- (Alt+F1) | **Software Engineering | Code Editor | Using Intellisense**
- (Alt+F1) | **Software Engineering | Code Editor | Browse to Source File**

10.8.1.2 General Workflow

In working with the Model Driven Development Environment, the build-debug workflow is a cyclic process. Beginning with configuring and setting up scripts, each iteration of the cycle can include:

- Model - Edit - Build - Debug - Test - Visual Analysis - Profile

When the cycle is complete, you document and deploy the solution.

Code Synchronization:

It is important that the model and source code are kept synchronized for the Visual Execution Analyzer to produce useful results; use the Code Generation tools to synchronize your model after any design changes or code editing.

Always build the application prior to any Execution Analysis session - debugging, recording or profiling.

Learn more

- [Managing Analyzer Scripts](#) ^[1826]
- [Build Application](#) ^[1866]
- [Update Package Contents](#) ^[1774]

10.8.1.3 *Workspace Layout and Tools*

This topic identifies the workspace layouts that you can select for debugging and Visual Execution Analysis, and the tools these layouts make available in the Model Driven Development Environment.

Workspace Layouts

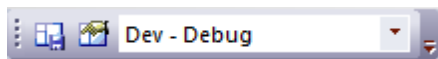
You can choose from many predefined workspace layouts, depending on the tasks you perform. When you are familiar with the environment and controls available to you, you can also define your own workspace layouts.

You can select from the following debug/execution analysis layouts:

- Dev - Debug
- VEA - Profile
- VEA - Record
- VEA - Simulate
- VEA - Test Points

You select the workspace layout from either the:

- Workspace Toolbar or
- **Workspace** field on the Start Page



Toolbar



Start Page

Tools

Whether you open the appropriate windows through a workspace layout or select them individually through their menu options, the following tools and facilities are of particular relevance to coding, building and debugging:

- Code engineering
- Source Code Editors
- Breakpoints and Markers
- Intellisense

- Execution Analyzer
- Debug window
- Locals window
- Call Stack window

Learn more

- [Workspace Layouts](#) ^[149]
- [Code Engineering](#) ^[1733]
- [Editing Source Code](#) ^[1799]
- [Intellisense](#) ^[1812]
- [Analyzer Scripts](#) ^[1827]
- [Build Application](#) ^[1866]
- [Breakpoint and Marker Management](#) ^[1869]
- [Debugger Facilities](#) ^[1874]

10.8.2 Analyzer Scripts

Analyzer Scripts are used to record information on compiling and running the application under development.

An Analyzer Script can be defined for a package-tree containing Classes with links to the code. The information you provide in a script controls the actions in, and provides support for:

- Building
- Debugging
- Testing
- Execution
- Deployment

Learn more

- [Analyzer Script Editor](#) ^[1831]
- [Managing Analyzer Scripts](#) ^[1826]
- [Add Build Commands](#) ^[1832]
- [Clean](#) ^[1833]
- [Debug](#) ^[1867]
- [Testing](#) ^[1834]
- [Add Run Command](#) ^[1862]
- [Add Deploy Command](#) ^[1862]
- [Configure Recording Detail](#) ^[1863]
- [Set Up Simulation Script](#) ^[2109]
- [Visual Execution Analyzer](#) ^[2143]

10.8.2.1 Managing Analyzer Scripts







The Execution Analyzer window enables you to manage all Analyzer scripts in the model. You use the window toolbar buttons or script context menu options to control script tasks. Scripts are listed by package; the list only shows packages that have Analyzer scripts defined against them.

Each user can activate one script at any time, independent of other users of the same model; one user activating a script does not impact the currently active scripts for other users or affect the scripts available to them.

Access **Analyzer | Execution Analyzer (Shift + F12)**

Toolbar Options:









Toolbar Button	Action	See also
	List the Analyzer windows; select and display the required window.	Debugging ^[1867] Breakpoint and Marker Management ^[1869] View the Call Stack ^[1884] View the Local Variables ^[1877] View Variables in Other Scopes ^[1880] Inspect Process Memory ^[1886] The Recording History ^[2149] The Testpoints Window ^[2189] Profiling Native Applications ^[2170] Model Simulation ^[2100]
	Add new Analyzer Script. A prompt displays to browse for and select the parent package, followed by the Analyzer Script Editor.	Analyzer Script Editor ^[1831]
	Export Scripts. You can export one or more Analyzer Scripts to an XML file, which can be used to import the scripts into another model. The Execution Analyzer: Export dialog displays, from which you select the script or scripts to export, followed by a prompt for the target file name and location.	
	Import Scripts.	

Toolbar Button	Action	See also
	<p>You can import one or more Analyzer Scripts from a previously exported XML file.</p> <p>The Browse Project dialog displays, on which you select the package into which to import the scripts, followed by a prompt for the source file name and location.</p>	
	Execute the Debug command of the active script.	Debug ^[1837] (scripts)
	Execute the Build command of the active script.	Add Commands ^[1832]
	Execute the Test command of the active script.	Add Testing Command ^[1834]
	Execute the Run command of the active script.	Add Run Command ^[1862]
	Execute the Deploy command of the active script.	Add Deploy Command ^[1862]
	Display the Help topic for this window.	

Context Menu Options:

Right-click on the required script or package to display the context menus.

Option	Action	See also
Add New Script	<p>Add a new script to the selected package.</p> <p>The Analyzer Scripts Editor displays, showing the Build page.</p>	Analyzer Script Editor ^[1837]
Paste Script	<p>Paste a copied script from the Enterprise Architect clipboard into the selected package.</p> <p>You can paste the copied script several times; each copy has the suffix Copy.</p> <p>To rename the copied script, press (F2) and overwrite the script name.</p>	
Export Scripts	<p>Export scripts from the selected package.</p> <p>The Execution Analyzer: Export dialog displays, from which you select the script or scripts to export, followed by a prompt for the target file name and location.</p>	

Option	Action	See also
Import Scripts	Import scripts from a .XML file into the selected package. A prompt displays for the source file name and location.	
Select In Project Browser	Highlight the selected package in the Project Browser. Display the Project Browser, which is now expanded to show the highlighted package.	
Build	Execute the Build command of the selected script.	Build Application 
Clean	Execute the Clean command of the selected script.	Clean 
Rebuild	Execute the Clean and Build commands of the selected script.	Build Application 
Debug	Execute the Debug command of the selected script.	Debug  (scripts)
Run	Execute the Run command of the selected script.	Add Run Command 
Test	Execute the Test command of the selected script.	Add Testing Command 
Deploy	Execute the Deploy command of the selected script.	Add Deploy Command 
Edit	Open the selected script in the Analyzer Scripts Editor.	Analyzer Script Editor 
Copy	Copy the selected script to the Enterprise Architect clipboard.	
Paste	Paste the most-recently copied script to the same package as the selected script. You can paste the copied script several times; each copy has the suffix Copy . To rename the copied script, press (F2) and overwrite the script name.	
Delete	Delete the selected script; there is no prompt for confirmation.	

Option	Action	See also
	To delete a <i>package</i> from the Execution Analyzer window, delete the scripts from the package; when the last script is deleted, the package is no longer listed.	
Set as Model Default	Set the selected script as the default script for the model. The icon to the left of the script changes color; any previously-set default script reverts to normal.	
Help	Display the Help topic for this window.	

Learning Center topics

- (Alt+F1) | **Build and Debug | Overview | Analyzer Scripts**

10.8.2.2 Analyzer Script Editor

Analyzer Scripts are associated with a Package. When you create an Analyzer Script you can define a number of actions - many features, such as Debugging, Testing, Profiling and Simulation, depend on and function according to how the active script has been configured.

The Execution Analyzer Script Editor has a number of pages for configuring various scripts; if you plan to use any of the features of the Execution Analyzer, you must complete at least the Build and Debug pages.

Task	Page	Actions	See also
Build	Build	Enter script or command to build the application.	Add Build Commands ^[1832]
	Clean	Enter script or command to clean the previous build .	Clean ^[1833]
Test	Test	Enter script or command to test the application.	Test ^[1834]
	Testpoints	Specify where the output from a Testpoint run is sent.	Testpoint Outputs ^[1836]
Debug	Platform	Specify the debugging platform, the application to be debugged, and the mode of debugging (attach to process or run).	Debug ^[1867]
	Tracepoints	Specify where the output from Tracepoints encountered during a debug session are sent.	Trace Output ^[1860]
	Workbench	Specify required information for using the Object Workbench control.	Workbench Setup ^[1861]
Run	Run	Enter a script or command to run the application.	Add Run Command ^[1862]
Deploy	Deploy	Enter a script or command to deploy the project.	Add Deploy Command ^[1862]
Recording	Recording	Exclude areas of the application from being recorded; you can	Configure

Task	Page	Actions	See also
g		specify wildcard expressions to exclude operations, Classes and modules.	Recording Detail ^[1863]
Simulation	Simulation	Complete the configuration for Simulation Control.	Model Simulation ^[2100]

10.8.2.3 Add Build Commands

This topic describes how you specify the commands to build the project or package, using the Execution Analyzer Script Editor.

The Build page enables you to enter commands to build your project. You can use Enterprise Architect Local Paths and environment variables in composing your command line(s).

Access **Analyzer | Execution Analyzer (Shift + F12) : New** (Toolbar icon) > **Build > Build Analyzer | Execution Analyzer (Shift + F12) : locate and double-click on required script > Build > Build**

Reference

On the Execution Analyzer Script Editor:

Field	Action	See also
Execute Command As:	<p>Batch File</p> <p>Use this option to enter multiple commands, which are executed as entered in a command window; the command window has access to your environment variables.</p> <p>Process</p> <p>Use this option to run a single program.</p> <p>The command should specify the path to the program, plus any command line arguments; if the executable path or any arguments contain spaces, they must be surrounded by quotes.</p>	Analyzer Script Editor ^[1837]
Build Script	<p>Write your script in the large text box, using the standard Windows Command Line commands; the format and content of this section depends on the actual compiler, make system, linker and so on that you use to build your project.</p> <p>You can specify, for example, compiler and linker options, and the names of output files; you can also wrap up all these commands into a convenient batch file and call that here instead.</p>	Local Paths ^[1900]

Field	Action	See also
	<p>When building using the Process option from above, you can add the %r token immediately preceding the files to be built. Enterprise Architect will recursively replace the token with the names of the child directories of the initial directory and execute the command again.</p> <p>Examples:</p> <p>Visual Studio:</p> <pre>" C: \ Program Files (x86) \ Mi crosof t Vi sual St udi o 9. 0 \ Com mon 7 \ I DE \ de ven v. com" / Re bui l d De bu g Re nt al Sys t em. sl n</pre> <p><i>Using a Local Path:</i></p> <pre>" %VsCompPat h% de ven v. exe" / bui l d De bu g Sub way. sl n</pre> <p>Java:</p> <pre>C: \ Program Files (x86) \ Java \ j dk1. 6. 0_22 \ bi n \ jav ac. exe" - g - cp "%cl asspat h% ." %r *. j ava</pre> <p><i>Using a Local Path:</i></p> <pre>" %JAVA% bi n \ jav ac. exe" - g - cp "%cl asspat h % ." %r *. j ava</pre>	
Default Directory	Type in or browse for the default directory path into which the Build commands are to load the built application files.	
Parse Output	<p>This enables you to select a method for automatically parsing the compiler output.</p> <p>If you select this option, output from the script is logged in the System Output window; Enterprise Architect parses the output according to the syntax you specify.</p>	The Output Window ^[164]

Learning Center topics

- (Alt+F1) | [Build and Debug | Overview | Build](#)
- (Alt+F1) | [Build and Debug | Build and Run | Build Scripts - Native](#)
- (Alt+F1) | [Build and Debug | Build and Run | Build Scripts - .NET](#)
- (Alt+F1) | [Build and Debug | Build and Run | Build Scripts - Java](#)

10.8.2.4 Clean

This topic explains how you enter a command to clean your project prior to a build; for example, to delete cached items used for building the project so that the next build recreates from scratch.

In Java you might delete all Class files, .jar files and .war files in the project; Visual Studio projects, however, might issue a call to the Visual Studio Clean operation instead.

When you execute the **Rebuild** menu option on a script, the command you specify here is executed followed by the Build command.

Access **Analyzer | Execution Analyzer (Shift + F12) : New** (Toolbar icon) > **Build > Clean**
Analyzer | Execution Analyzer (Shift + F12) : locate and double-click on required script > Build > Clean

Topics

Topic	Detail	See also
Usage	Enter the command to be executed when you select Clean from the script context menu.	
Example	<code>devenv.com / Clean Debug MyProject.sln</code>	

Learn more

- [Add Build Commands](#)^[1832]

10.8.2.5 Testing

The Execution Analyzer supports testing using xUnit testing applications and Testpoints.

This section describes how to create commands for performing unit testing and Testpoint testing on your code.

Learn more

- [Add Testing Command](#)^[1834]
- [Testpoints Output](#)^[1836]

Learning Center topics

- **(Alt + F1) | Execution Analysis | xUnit Testing**
- **(Alt + F1) | Execution Analysis | Testpoint Testing**

10.8.2.5.1 Add Testing Command

This topic explains how to configure the Test Page of an Analyzer Script for performing unit testing on your code.

In order for this to be possible, you must first create a Build script for the appropriate package.

Access **Analyzer | Execution Analyzer (Shift + F12) : New** (Toolbar icon) > **Test > Test**
Analyzer | Execution Analyzer (Shift + F12) : locate and double-click on required script > Test > Test

Topics

Topic	Detail	See also
Usage	<p>A sample script would contain a line to execute the testing tool of your choice, with the filename of the executable produced by the Build command as the option.</p> <p>Testing could be integrated with any test tool using the command line provided, but in these examples you can see how to integrate <i>NUnit</i> and <i>JUnit</i> testing with your source code.</p> <p>Enterprise Architect provides inbuilt MDA Transforms from source to Test Case, plus the ability to capture <i>xUnit</i> output and use it to go directly to a test failure; <i>xUnit</i> integration with your model is a powerful means of delivering solid and well-tested code as part of the complete model-build-test-execute-deploy life-cycle.</p> <p>NUnit and JUnit must be downloaded and installed prior to their use; Enterprise Architect does not include these products in the base installer.</p>	<p>Managing Analyzer Scripts [1828]</p> <p>Add Build Commands [1832]</p> <p>JUnit Transformation [1702]</p> <p>NUnit Transformation [1704]</p> <p>Unit Testing [2183]</p>
Execute Command As:	<p>Process</p> <p>Enter the path to a program or batch file to run followed by any parameters.</p> <p>Batch File</p> <p>When using this option you can enter multiple commands, which are then executed as a single script in a command console; you have access to any environment variables available in a standard command console.</p>	
Example	<p>NUnit</p> <pre>" C:\Program Files\NUnit\bin\nunit-console.exe "pin\debug\Calculator.exe"</pre> <p>JUnit</p> <pre>java junit.textui.Testrunner %N</pre> <p>The command listed in this field is executed as if from the command prompt; as a result, if the executable path or any arguments contain spaces, they must be surrounded in quotes.</p>	
Default Directory	Preset to the Build default directory.	Add Commands [1832]
Parse Output	<p>Selecting an Output Parser causes output from the command(s) to be captured to the Output window; the Output Parser field specifies the syntax to be expected.</p> <p>Double-clicking on a result in the Output window opens the corresponding code segment in Enterprise Architect's code</p>	Locate Compiler

Topic	Detail	See also
	window.	Errors in Code <small>[1866]</small>
Build First	Select to ensure that the package is recompiled each time you run the test. If you include the string %N in your test script it is replaced by the fully namespace-qualified name of the currently selected Class when the script is executed.	

Learning Center topics

- (Alt + F1) | **Build and Debug | Test**

10.8.2.5.2 Testpoints Output

The Testpoints page of the Analyzer Script allows you to configure the output of a Testpoint run.

Access **Analyzer | Execution Analyzer (Shift + F12) : New** (Toolbar icon) > **Test > Testpoints**
Analyzer | Execution Analyzer (Shift + F12) : locate and double-click on required script > Test > Testpoints

Topic	Detail	See also
Output	You can select from two options: <ul style="list-style-type: none"> • Screen (Default) - The output is directed to the Testpoints tab of the System Output window • File - The output is directed to file 	Testpoint Management <small>[2187]</small>
Folder	Enter the folder to use for Testpoint log files.	
Filename	Enter the name to use for the Testpoints log files.	
Overwrite	When this option is selected, the file specified is overwritten each time a Testpoint run is performed.	
Auto Number	When this option is selected, the Testpoint output is composed of the filename you specify and the number of the Test run; each time you perform a Test run the number is incremented.	
Prefix trace output with function	When this option is selected, any trace statements executed during the Testpoint run are prefixed with the current function call.	

Topic	Detail	See also

Learning Center topics

- [\(Alt + F1\) | Execution Analysis | Testpoints | Introducing Testpoints](#)

10.8.2.6 Debug

To set up a debug script you must specify:

- The debugger to use
- The application path
- Runtime options, if applicable

You specify these on the Debug > Platform page of the Analyzer Script dialog.

Access [Analyzer | Execution Analyzer \(Shift + F12\) : New](#) (Toolbar icon) > [Debug > Platform Analyzer | Execution Analyzer \(Shift + F12\)](#) : locate and double-click on required script > [Debug > Platform](#)

Learn more

- [Operating System Specific Requirements](#) ¹⁸³⁷

Learning Center topics

- [\(Alt+F1\) | Build and Debug | Overview | Debug](#)

10.8.2.6.1 Operating System Specific Requirements

Important - Please read the information provided in this topic.

Topics

Topic	Detail	See also
Prerequisites	Creation of an Analyzer script and configuration of the Debug command in that script.	
Supported Platforms	Enterprise Architect supports debugging on these platforms: .NET <ul style="list-style-type: none"> • Microsoft™ .NET Framework 1.1 and later, including .NET 4 • Language support: C, C#, C++, J#, Visual Basic Java <ul style="list-style-type: none"> • Java 2 Platform Standard edition (J2SE) version 5.0 	.NET ¹⁸⁵⁰ Java ¹⁸⁴³

Topic	Detail	See also
	<ul style="list-style-type: none"> • J2EE JDK 1.4 and above • Requires previous installation of the Java Runtime Environment and Java Development Kit from Sun Microsystems™ <p>Debugging is implemented through the Java Virtual Machine Tools Interface (JVMTI), which is part of the Java Platform Debugger Architecture (JPDA); the JPDA is included in the J2SE SDK 1.3 and later.</p> <p>Windows for Native Applications</p> <p>Enterprise Architect supports debugging native code (C, C++ and Visual Basic) compiled with the Microsoft™ compiler where an associated PDB file is available.</p> <p>You can import native code into your model, and record the execution history for any Classes and methods.</p> <p>You can also generate Sequence diagrams from the resulting execution path.</p> <p>PHP</p> <p>Enterprise Architect enables you to perform local and remote debugging with PHP.exe.</p> <p>GNU Debugger (GDB)</p> <p>Enterprise Architect supports debugging using the GNU Debugger, which enables you to debug your applications under Linux either locally or remotely.</p>	<p>Microsoft C++ and Native (C, VB)¹⁸⁴¹</p> <p>The PHP Debugger¹⁸⁵⁴</p> <p>The GNU Debugger¹⁸⁵⁹</p>

Notes

- Debugging under Windows Vista (x64) - if you encounter problems debugging with Enterprise Architect on a 64-bit platform, you should build a Win32 platform configuration in Visual Studio; that is, do not specify **ANY-CPU**, specify **WIN32**

10.8.2.6.1.1 UAC-Enabled Operating Systems

The Microsoft operating systems *Windows Vista* and *Windows 7* provide User Account Control (UAC) to manage security for applications.

The Enterprise Architect Visual Execution Analyser is UAC-compliant, and users of UAC-enabled systems can perform operations with the Visual Execution Analyser and related facilities under accounts that are members of only the *Users* group.

However, when attaching to processes running as services on a UAC-enabled operating system, it might be necessary to log in as an Administrator.

How to

To log in as an Administrator, follow the step below:

Step	Action	See also
1	Before you run Enterprise Architect, right-click on the Enterprise Architect icon on the desktop and select the Run as administrator option.	



Alternatively, to edit or create a link to Enterprise Architect and configure the link to run as an Administrator

Step	Action	See also
1	Right-click on the Enterprise Architect icon and select the Properties menu option. The Enterprise Architect Properties dialog displays.	
2	Click on the Advanced button. The Advanced Properties dialog displays.	
3	Select the Run as administrator checkbox.	
4	Click on the OK button, and again on the Enterprise Architect Properties dialog.	

10.8.2.6.1.2 WINE Debugging**How to**

To configure Enterprise Architect to debug under WINE

Step	Action	See also
1	At the command line, run <code>\$ winecfg</code> .	
2	Set the library overrides for <code>dbghelp</code> to (native, builtin) , and accept the warning about overriding this DLL.	
3	Set <code>dbghelp</code> to native by using <code>winecfg</code> .	
4	Copy the application source code plus executable(s) to your bottle.	

Step	Action	See also
	The path must be the same as the compiled version; that is: If Windows source = C: \ Sour ce\ Samp l eApp, under Crossover it must be C: \ Sour ce\ Samp l eApp	
5	Copy any Side-By-Side assemblies that are used by the application.	
6	Import the source code into Enterprise Architect (optional).	
7	Create a build script on a package. Set the path of the application on the Debug tab, and set the Use Debugger field to Microsoft Native .	Add Build Commands 
8	Open the Profiler (Analyzer Profile).	Visual Execution Profiler 
9	Click on the Launch button (first button on the Profiler window).	
10	If the sample didn't start, click on the Sampling button (third button on the Profiler window).	
11	Once you have finished profiling, shut down the application (<i>not</i> Enterprise Architect).	
12	View the Sampler report by clicking the View Report button (fifth button on the Profiler window).	

Topics

Topic	Detail	See also
Access Violation Exceptions	<p>Due to the manner in which WINE handles direct drawing and access to DIB data, an additional option is provided on the Debug window toolbar drop-down menu to ignore or process access violation exceptions thrown when your program directly accesses DIB data.</p> <p>Select this option to catch genuine (unexpected) access violations; deselect it to ignore expected violations.</p> <p>As the debugger cannot distinguish between expected and unexpected violations, you might have to use trial and error to capture and inspect genuine program crashes.</p>	

Notes

- If WINE crashes, the back traces might not be correct
- If you are using MFC remember to copy the debug side-by-side assemblies to the *C:\window\winsxs* directory
- To add a windows path to WINE, modify the Registry entry:

HKEY_LOCAL_MACHINE\System\Current Control Set\Control\Session Manager\Environment

10.8.2.6.2 Microsoft C++ and Native (C, VB)**Topics**

Topic	Detail	See also
Usage	<p>You can debug native code only if there is a corresponding PDB file for the executable; you normally create the PDB file as a result of building the application.</p> <p>The build should include full debug information and there should be no optimizations set.</p> <p>The script must specify two things to support debugging:</p> <ul style="list-style-type: none"> • The path to the executable • Microsoft Native as the debugging platform 	General Setup ¹⁸⁴¹

Learning Center topics

- (Alt+F1) | **Build and Debug | Debug | Microsoft Native**

10.8.2.6.2.1 General Setup

This is the general setup for debugging Microsoft Native Applications (C++, C, Visual Basic). You have two options when debugging:


- Debug an application
- Attach to an application that is running

Option 1 - Debug an application

Field	Action	See also
Debugger	Select Microsoft Native as the debugging platform.	
Mode	Select the Run radio button.	

Field	Action	See also
Default Directory	This is set as the default directory for the process being debugged.	
Application Path	Select and enter either the full or the relative path to the application executable.	
Command Line Arguments	Parameters to pass to the application at startup	
Show Console	Create a console window for the debugger; not applicable for attaching to a process.	
Symbol Search Paths	Specify any additional paths to locate debug symbols for the debugger; separate the paths with a semi-colon.	

Option 2 - Attach to an application that is running

Field	Action	See also
Debugger	Select Microsoft .NET as the debugging platform.	
Mode	Select the Attach to Process radio button.	Debug Another Process 
Symbol Search Paths	Specify any additional paths to locate debug symbols for the debugger. You could specify a symbol server here if you prefer; separate the paths with a semi-colon or comma.	

10.8.2.6.2.2 *Debug Symbols*

Topics

Topic	Detail	See also
Abstract	<p>For applications built using <i>Microsoft Platform SDK</i>, Debug Symbols are written to an application PDB file when the application is built.</p> <p>The <i>Debugging Tools for Windows</i>, an API used by the Visual Execution Debugger, uses these symbols to present meaningful information to Execution Analyzer controls.</p> <p>These symbols can easily get out of date and cause errant behaviour - the debugger might highlight the wrong line of code in the editor whilst at a breakpoint; it is therefore best to ensure the application is built prior to any debugging or recording session.</p>	

Topic	Detail	See also
	<p>The debugger must inform the API how to reconcile addresses in the image being debugged; it does this by specifying a number of paths to the API that tell it where to look for PDB files.</p> <p>For system DLLs (<i>kernel32</i>, <i>mfc90ud</i>) for which no debug symbols are found, the Call Stack shows some frames with module names and addresses only.</p> <p>You can supplement the symbols translated by passing additional paths to the API; you pass additional symbol paths in a semi-colon separated list in the Debug tab.</p>	

10.8.2.6.3 Java

This section describes how to set up Enterprise Architect for debugging Java applications and Web Servers.

Learn more

- [General Setup for Java](#)^[1843]
- [Advanced Techniques](#)^[1845]
- [Working with Java Web Servers](#)^[1846]

10.8.2.6.3.1 General Setup for Java

This is the general setup for debugging Java Applications. You have two options when debugging.

- Debug an Application
- Attach to an application that is running

Option 1 - Debug an Application

Field	Action	See also
Debugger	Select Java .	
Mode	Select Run .	
Default Directory	This path is added to the class path property when the Java Virtual Machine is created.	
Application Class	<p>Identify the fully qualified Class name to debug; the Class must have a method declared with the following signature:</p> <pre>public static void main(String());</pre> <p>Application Class <input type="text" value="samples.Collector"/></p> <p>Command Line Arguments: </p>	

Field	Action	See also
Command Line Arguments	<p>Specify any parameters to be passed to the main method of the Application Class above.</p> <p>Parameters containing spaces should be surrounded with double quotes.</p>	
Java Virtual Machine Options	<p>Specify command line options for Virtual Machine creation.</p> <p>You also must provide a parameter (jre) that is a path to be searched for the jvm.dll; this is the DLL supplied as part of the Java runtime environment or Java JDK from Sun Microsystems™.</p> <p>Java Virtual Machine Options:</p> <pre>jre=%JAVA%\jre,-Djava.class.path=%classpath%;;</pre> <p>In the example above, a virtual machine is created with a new Class path property that comprises any paths named in the <i>CLASSPATH</i> environment variable plus the default directory.</p> <p>If no Class path is specified, the debugger always creates the virtual machine with a Class path property equal to any path contained in the environment variable plus the path entered in the default working directory of this script.</p> <p>If source files and .class files are located under different directory trees, the Class path property MUST include both root path(s) to the source and root path(s) to binary class files.</p>	

Option 2 - Attach to Virtual Machine

There is very little to specify when attaching to a VM; however, the VM must have the Sparx Systems debugging agent loaded.

Field	Action	See also
Debugger	Select Java	Attach to Virtual Machine ^[1845]
Mode	Select Attach to Virtual Machine	Debug Tomcat Server ^[1849] Debug Another Process ^[1888]

Learning Center topics

- (Alt+F1) | **Build and Debug | Debug | Java**

10.8.2.6.3.2 Advanced Techniques

In addition to the standard Java debugging techniques, you can:

- [Attach to a Virtual Machine](#)^[1845]
- [Debug Internet Browser Java Applets](#)^[1845]

Topics

Topic	Detail	See also
Abstract	<p>You can debug a Java application by attaching to process that is hosting a Java Virtual Machine; you might want to do this for attaching to a webserver such as Tomcat or JBOSS.</p> <p>The <i>Java Virtual Machine Tools Interface</i> from Sun Microsystems, is the API used by Enterprise Architect; it allows a debugging agent to be specified when the JVM is created.</p> <p>To debug a running JVM from Enterprise Architect, the Sparx Systems' debugging agent must have been specified as a startup option to the JVM when it was started; how this accomplished for products such as Tomcat and JBOSS should be researched from that product's own documentation.</p> <p>For <i>java.exe</i>, the command line option to load the Enterprise Architect debugging agent is:</p> <pre>- agentlib: SSJavaProfiler75</pre> <p>or:</p> <pre>- agentpath: "c:\program files\sparx systems\ea\SSJavaProfiler75"</pre> <p>It is not necessary to configure an Analyzer Script when you attach to a Virtual Machine; you can just use the Attach button on one of the Analyzer toolbars.</p> <p>If you configure an Analyzer Script, there are only two things that must be selected:</p> <ul style="list-style-type: none"> • Select Java as the debugging platform • Choose the Attach to Virtual Machine option 	<p>Debug Another Process^[1888]</p>

This topic describes the configuration requirements and procedure for debugging Java Applets running in a browser from Enterprise Architect.

How to

To attach to the browser process hosting the Java Virtual Machine (JVM) from Enterprise Architect

Step	Action	See also
1	Ensure binaries for the applet code to be debugged have been built with debug information.	
2	Configure the JVM using the Java Control Panel.	

Step	Action	See also
3	In the Java Applet Runtime Settings panel, click on the View button.	
4	On the installed version to use, include the following option in the Runtime Parameters field: <pre>- agent path : c : \ p r o g r a m f i l e s \ s p a r x s y s t e m s \ e a \ S S J a v a P r o f i l e r 7 5</pre>	
5	In this field add the required Class paths. At least one of these paths should include the root path of the source files to use in debugging.	
6	Set breakpoints.	Breakpoint and Marker Management ¹⁸⁶⁹
7	Launch the browser.	
8	Attach to the browser process from Enterprise Architect.	

10.8.2.6.3.3 Working with Java Web Servers

This topic describes the configuration requirements and procedure for debugging Java web servers such as JBOSS and Apache Tomcat (both Server configuration and Windows Service configuration) in Enterprise Architect.

How to

To attach to the process hosting the Java Virtual Machine from Enterprise Architect

Step	Action	See also
1	Ensure binaries for the web server code to be debugged have been built with debug information.	
2	Launch the server with the Virtual Machine startup option described in <i>Server Configuration</i> , below.	
3	Import source code into the Enterprise Architect Model, or synchronize existing code.	
4	Set breakpoints.	Breakpoint and Marker Management ¹⁸⁶⁹
5	Launch the client.	

Step	Action	See also
6	Attach to the process from Enterprise Architect.	

Topics

Topic	Detail	See also
Server Configuration	<p>The configuration necessary for the web servers to interact with Enterprise Architect must address the following two essential points:</p> <ul style="list-style-type: none"> Any VM to be debugged, created or hosted by the server must have the Sparx Systems Agent <code>SSJavaPr of i l er 75</code> command line option specified in the VM startup option (that is: <code>- agent l i b: SSJavaPr of i l er 75</code>) The CLASSPATH, however it is passed to the VM, must specify the root path to the package source files <p>The Enterprise Architect debugger uses the <code>java.class.path</code> property in the VM being debugged, to locate the source file corresponding to a breakpoint occurring in a Class during execution; for example, a Class to be debugged is called:</p> <p style="padding-left: 20px;"><code>a . b . C</code></p> <p>This is located in physical directory:</p> <p style="padding-left: 20px;"><code>C : \ s o u r c e \ a \ b</code></p> <p>So, for debugging to be successful, the CLASSPATH must contain the root path:</p> <p style="padding-left: 20px;"><code>c : \ s o u r c e</code></p>	
Analyzer Script Configuration	<p>Using the Debug tab of the Build Script dialog, create a script for the code you have imported and:</p> <ul style="list-style-type: none"> Select the Attach to process radio button and, in the field below, type attach In the Use Debugger field, click on the drop-down arrow and select Java <p>All other fields are unimportant; the Directory field is normally used in the absence of any Class path property.</p>	General Setup for Java ^[1843]

How to

To debug

Step	Action	See also
1	Ensure that the server is running, and that the server process has loaded the Sparx Systems Agent. DLL SSJavaPr of i l e r 75. DLL (Use <i>Process Explorer</i> or similar tools to prove this).	
2	In Enterprise Architect, open the source code and set some breakpoints.	Breakpoint and Marker Management ^[1869]
3	Click on the Run Debug button in Enterprise Architect. The Attach To Process dialog displays.	Debug Another Process ^[1888]
4	Select the server process hosting the application.	
5	Click on the OK button. A confirmation message displays in the Debug window, stating that the process has been attached.	

The breakpoints could show a question mark. In this case the Class might not have been loaded yet by the VM. If the question mark remains even after you are sure the Class containing the breakpoint has been loaded, then either:

- The binaries being executed by the server are not based on the source code
- The debugger cannot reconcile the breakpoint to a source file (check class paths), or
- The JVM has not loaded the Sparx Systems agent

Learn more

- [JBoss Server](#)^[1848]
- [Apache Tomcat Server](#)^[1849]
- [Apache Tomcat Windows Service](#)^[1850]

Topics

Topic	Detail	See also
Abstract	Consider the JBoss example below; the source code for a simple servlet is located in the directory location: <i>C:\Benchmark\Java\JBoss\Inventory</i> The binaries executed by JBoss are located in the JAW.EAR file in this location: <i>C:\JBoss\03b-dao\build\distribution</i>	

Topic	Detail	See also
	<p>The Enterprise Architect debugger has to be able to locate source files during debugging; to do this it also uses the CLASSPATH, searching in any listed path for a matching JAVA source file, so the CLASSPATH must include a path to the root of the package for Enterprise Architect to find the source during debugging.</p> <p>The following is an excerpt from the command file that executes the JBOSS server; the Class to be debugged is at:</p> <pre>com/inventory/dto/car DTO</pre> <p>Therefore, the root of this path is included in the <i>JBOSS</i> classpath</p>	
Example Code	<pre>RUN. BAT ----- set SOURCE=C: \ Benchmark\ Java\ JBOSS\ I nvent or y set JAVAC_JAR=%JAVA_HOME%\ lib\ tool s. jar if "%JBOSS_CLASSPATH%" == "" (set JBOSS_CLASSPATH=%SOURCE% % JAVAC_JAR% %RUNJAR%) el se (set JBOSS_CLASSPATH=%SOURCE% % JBOSS_CLASSPATH% %JAVAC_JAR% %RUNJAR%) set JAVA_OPTS=%JAVA_OPTS% - agent path: " c: \ program files\ spar x system\ ssj avapr of il er 75"</pre>	

Topics

Topic	Detail	See also
Configurati on	<p>This configuration is for the same application as outlined in the <i>JBOSS server</i> configuration topic</p> <p>The following two points are important:</p> <ul style="list-style-type: none"> The Java VM option: <pre>- agent path: c:\ program files\ spar x system\ ea\ ssj avapr of il er 75</pre> The addition to the Class path property of the path to the source code: <pre>C:\ JBOSS\ 03b- dao\ common\ src;</pre> 	JBOSS Server ^[1848]

Topic	Detail	See also
	<input checked="" type="checkbox"/> Use default Java Virtual Machine: <input type="text" value="C:\SunJava\jre 1.7.0\bin\client\jvm.dll"/> ... Java Classpath: <input type="text" value="C:\Tomcat Local Source\src;C:\Java\jdk1.6.0_07\jre;C:\Java\jdk1.6.0_07\bin"/> ... Java Options: <input type="text" value="-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.util.logging.config.file=C:\Program Files\Apache Software Foundation\Tomcat 7.0\conf\tomcat7.0\logging.properties -agentpath:C:\program files\sparx systems\ea\ssjavaprofiler75"/> ... <p style="text-align: center;">Tomcat Server Configuration Example</p>	

Topics

Topic	Detail	See also
Configuration	<p>For users running Apache Tomcat as a Windows™ service, it is important to configure the service to enable interaction with the Desktop; failure to do so causes debugging to fail within Enterprise Architect.</p> <p>Log on as:</p> <p><input checked="" type="radio"/> Local System account</p> <p><input checked="" type="checkbox"/> Allow service to interact with desktop</p> <p>Select the Allow service to interact with desktop checkbox.</p>	

10.8.2.6.4 .NET

This section describes how to configure Enterprise Architect for debugging .NET applications. It covers:

- [General Setup](#)^[1850]
- [Debugging Managed code in an Unmanaged Application](#)^[1851]
- [Debug COM Interop](#)^[1852]
- [Debug ASP .NET](#)^[1853]

10.8.2.6.4.1 General Setup for .NET

This is the general setup for debugging Microsoft .NET applications. You have two options when debugging:

- Debug an Application
- Attach to an application that is running

Option 1 -Debug an application

Field	Action	See also
Debugger	Select Microsoft .NET as the debugging platform.	Debugging an Unmanaged Application ^[1851]
Mode	Select the Run radio button.	Debug ASP.NET ^[1853]
Default Directory	This is set as the default directory for the process being debugged.	
Application Path	Select and enter either the full or the relative path to the application executable.	
Command Line Arguments	Parameters to pass to the application at startup.	
Show Console	Create a console window for the debugger; not applicable to attaching to a process.	
Symbol Search Paths	Specify any additional paths to locate debug symbols for the debugger; separate the paths with a semi-colon.	

Option 2 - Attach to an application that is running

Field	Action	See also
Debugger	Select Microsoft .NET as the debugging platform.	Debugging an Unmanaged Application ^[1851]
Mode	Select the Attach to Process radio button.	Debug ASP.NET ^[1853] Debug Another Process ^[1888]

Learning Center topics

- (Alt+F1) | [Build and Debug | Debug | Microsoft .NET](#)

10.8.2.6.4.2 Debugging an Unmanaged Application

Topics

Topic	Detail	See also
Usage	If you are debugging managed code using an unmanaged application, the debugger might fail to detect the correct version of the Common Language Runtime (CLR) to load.	http://www.msdn2.microsoft.com/en-us/library/9w519wzk.aspx


Topic	Detail	See also
	<p>You should specify a config file if you don't already have one for the debug application specified in the Debug command of your script.</p> <p>The config file should reside in the same directory as your application, and take the format:</p> <pre>name.exe.config</pre> <p>where <i>name</i> is the name of your application.</p> <p>The version of the CLR you specify should match the version loaded by the managed code invoked by the debuggee.</p> <p>The following is a sample config file:</p> <pre><configuration> <startup> <requiredRuntime version="version" /> </startup> </configuration></pre> <p><i>Version</i> is the version of the CLR targeted by your plugin or COM code.</p>	(Online Resource)

10.8.2.6.4.3 Debug COM Interop

Enterprise Architect enables you to debug .NET managed code executed using COM in either a Local or an In-Process server.

This feature is useful for debugging Plugins and ActiveX components.

How to

Step	Action	See also
1	Create a package in Enterprise Architect and import the code to debug.	Software Engineering 
2	Ensure the COM component is built with debug information.	
3	Create a Script for the Package.	
4	In the Debug Platform page, you can select to either attach to an unmanaged process or specify the path to an unmanaged application to call your managed code.	
5	Add breakpoints in the source code to debug.	

Topics

Topic	Detail	See also
Attach to an Unmanaged Process	<p>If you are using:</p> <ul style="list-style-type: none"> • An In-Process COM server, attach to the client process • A Local COM Server, attach to the server process <p>Click on the Debug window Run button (or press (F6)) to display a list of processes from which you can choose.</p>	Debug Another Process ^[1888]

Notes

- Detaching from a COM interop process you have been debugging terminates the process; this is a known issue for Microsoft .NET Framework, and information on it can be found on many of the MSDN .NET blogs

10.8.2.6.4.4 Debug ASP .NETTopics

Topic	Detail	See also
Usage	<p>Debugging for web services such as ASP requires that the Enterprise Architect debugger is able to attach to a running service.</p> <p>Begin by ensuring that the directory containing the ASP .NET service project has been imported into Enterprise Architect and, if required, the web folder containing the client web pages.</p> <p>If your web project directory resides under the website hosting directory, you can import from the root and include both ASP code and web pages at the same time.</p> <p>It is necessary to launch the client first, as the ASP .NET service process might not already be running; load the client using your browser - this ensures that the web server is running.</p> <p>In the debug setup you must select the Attach to process radio button, and then type the keyword Attach in the script; this keyword causes the debugger to prompt you for a process at runtime.</p> <p>Click on the Debug window Run button to start the debugger; the Attach To Process dialog displays.</p> <p>The name of the process varies across Microsoft operating systems, as explained in the <i>ASP .NET SDK</i>; for example, under Windows Vista the name of the IIS process is <i>w3wp.exe</i>.</p> <p>On Windows XP, the name of the process resembles <i>aspnet_wp.exe</i>, although the name could reflect the</p>	Run the Debugger ^[1875] Breakpoint and Marker Management ^[1869] Debug Another Process ^[1888]

Topic	Detail	See also
	<p>version of the .NET framework that it is supporting.</p> <p>There can be multiple ASP.NET processes running under XP; you must ensure that you attach to the correct version, which would be the one hosting the .NET framework version that your application runs on; check the <i>web.config</i> file for your web service to verify the version of .NET framework it is tied to.</p> <p>The Debug window Stop button should be enabled and any breakpoints should be red, indicating they have been bound.</p>	

You can set breakpoints at anytime in the web server code. You can also set breakpoints in the ASP web page(s) if you imported them.

Notes

- Some breakpoints might not have bound successfully, but if none at all are bound (indicated by being dark red with question marks) something has gone out of sync; try rebuilding and re-importing source code

10.8.2.6.5 The PHP Debugger

The Enterprise Architect PHP Debugger enables you to debug PHP.exe scripts. This section discusses basic setup and the various debugging scenarios that are commonly encountered; the scenarios concern themselves with the mapping of file paths, which is critical to the success of a remote debugging session.

- Script Setup
- Local Windows Machine (Apache Server)
- Local Windows Machine (PHP.exe)
- Remote Linux Machine (Apache Server)
- Remote Linux Machine (PHP.exe)

Topics

Topic	Detail	See also									
Script Setup	<p>An Analyzer Script is a basic requirement for debugging in Enterprise Architect; you create a script using the toolbar of the Execution Analyzer.</p> <p>Select PHPXDebug as the debugging platform; when you select this platform the property page displays the following connection settings:</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>port</td> <td>9000</td> <td>The port that Enterprise Architect listens on for the</td> </tr> <tr> <td>host</td> <td>localhost</td> <td>The adaptor that Enterprise Architect listens on for</td> </tr> </tbody> </table>	Field	Value	Meaning	port	9000	The port that Enterprise Architect listens on for the	host	localhost	The adaptor that Enterprise Architect listens on for	
Field	Value	Meaning									
port	9000	The port that Enterprise Architect listens on for the									
host	localhost	The adaptor that Enterprise Architect listens on for									

Topic	Detail	See also												
	<table border="1"> <thead> <tr> <th data-bbox="419 309 579 365">Field</th> <th data-bbox="579 309 746 365">Value</th> <th data-bbox="746 309 1281 365">Meaning</th> </tr> </thead> <tbody> <tr> <td data-bbox="419 365 579 477"></td> <td data-bbox="579 365 746 477"></td> <td data-bbox="746 365 1281 477">from PHP.</td> </tr> <tr> <td data-bbox="419 477 579 667">localpath</td> <td data-bbox="579 477 746 667">%LOCAL%</td> <td data-bbox="746 477 1281 667">Specifies the local file path to be mapped to a remote file path. This is a remote debugging setting; for local debugging the value is a placeholder; you should edit it to fit the local environment.</td> </tr> <tr> <td data-bbox="419 667 579 857">remotepath</td> <td data-bbox="579 667 746 857">%REMOTE%</td> <td data-bbox="746 667 1281 857">Specifies the remote file path that a local file path is mapped to. This is a remote debugging setting; for local debugging the value is a placeholder; you should edit it to fit the remote environment.</td> </tr> </tbody> </table>	Field	Value	Meaning			from PHP.	localpath	%LOCAL%	Specifies the local file path to be mapped to a remote file path. This is a remote debugging setting; for local debugging the value is a placeholder; you should edit it to fit the local environment.	remotepath	%REMOTE%	Specifies the remote file path that a local file path is mapped to. This is a remote debugging setting; for local debugging the value is a placeholder; you should edit it to fit the remote environment.	
Field	Value	Meaning												
		from PHP.												
localpath	%LOCAL%	Specifies the local file path to be mapped to a remote file path. This is a remote debugging setting; for local debugging the value is a placeholder; you should edit it to fit the local environment.												
remotepath	%REMOTE%	Specifies the remote file path that a local file path is mapped to. This is a remote debugging setting; for local debugging the value is a placeholder; you should edit it to fit the remote environment.												
Local Machine Apache Server	<p>In this situation, consider the following configuration:</p> <ul style="list-style-type: none"> • O/S: Windows 7 • Network computer name: <i>MyPC</i> • Network share MyShare mapped to <i>c:\myshare</i> • Source files in Enterprise Architect have been imported from <i>c:\myshare\apache\myapp\scripts</i> • Apache document root is set to <i>//MyPC/MyShare/apache</i> <p>In this scenario an Analyzer Script for the connection parameters might be configured as follows:</p> <ul style="list-style-type: none"> • host: localhost • port: 9000 • localpath: <i>c:\myshare\apache\</i> • remotepath: <i>MyPC/MyShare/apache/</i> 													
Local Machine PHP.EXE	<p>In this scenario an Analyzer Script for the connection parameters might be configured as follows, as file paths always map to same physical path:</p> <ul style="list-style-type: none"> • host: localhost • port: 9000 • localpath: • remotepath: 													
Remote	<p>In this situation consider the following configuration:</p>													

Topic	Detail	See also
Linux Machine Apache Server	<ul style="list-style-type: none"> • Local Machine <ul style="list-style-type: none"> • O/S: Windows7 • Source files in Enterprise Architect have been imported from c: <i>myshare\apache\myapp\scripts</i> • Remote Machine <ul style="list-style-type: none"> • O/S: Linux • Apache document root is set to <i>home/apache/htdocs</i> • Source files in Apache are located at <i>home/apache/htdocs/myapp/scripts</i> <p>In this scenario an Analyzer Script for the connection parameters might be configured as follows:</p> <ul style="list-style-type: none"> • host: localhost • port: 9000 • localpath: <i>c:\myshare\apache\</i> • remotepath: <i>home/apache/htdocs/</i> 	
Remote Linux Machine PHP.exe	<p>In this situation consider the following configuration:</p> <ul style="list-style-type: none"> • Local Machine <ul style="list-style-type: none"> • O/S: Windows7 • Source files in Enterprise Architect have been imported from c: <i>myshare\apache\myapp\scripts</i> • Remote Machine <ul style="list-style-type: none"> • O/S: Linux • Source files in Apache located at <i>home/myapp/scripts</i> <p>In this scenario an Analyzer Script for the connection parameters might be configured as follows:</p> <ul style="list-style-type: none"> • host: localhost • port: 9000 • localpath: <i>c:\myshare\apache\</i> • remotepath: <i>home/</i> 	

Learn more

- [PHP Debugger - System Requirements](#)¹⁸⁵⁷
- [PHP Debugger Checklist](#)¹⁸⁵⁷

Learning Center topics

- (Alt+F1) | **Build and Debug | Debug | PHP**
- (Alt+F1) | **Build and Debug | Debug | PHP Samples**

10.8.2.6.5.1 PHP Debugger - System Requirements

This topic identifies the system requirements and operating systems for the Enterprise Architect PHP debugger.

System Requirements:

- Enterprise Architect version 9
- PHP version 5.3 or above
- PHP zend extension XDebug 2.1 or above
- For web servers such as Apache, a server version that supports the PHP version

Supported Operating Systems:

- Client (Enterprise Architect)
 - Microsoft Windows XP and above
 - Linux running Crossover Office
- Server (PHP)
 - Microsoft Windows XP and above
 - Linux

10.8.2.6.5.2 PHP Debugger Checklist

This topic provides a supplementary checklist and troubleshooting guide for debugging PHP scripts in Enterprise Architect.

Topics

Topic	Detail	See also
System Requirements	<ul style="list-style-type: none"> • Apache HTTP Web Server version 2.2 • PHP version 5.3 or above • XDebug version 2.1.1 	http://httpd.apache.org/ http://windows.php.net/ http://www.xdebug.org/download.php
Checklist	Enterprise Architect <ul style="list-style-type: none"> • The model has an Analyzer Script configured to use the PHP XDebug platform • PHP source code has been imported into the model (for recording and testpoints) • When the PHP XDebug platform is selected from the Analyzer Script dialog, default runtime settings 	

Topic	Detail	See also
	<p>are listed in the Connection field</p> <ul style="list-style-type: none"> • <code>local path: %LOCAL%</code> • <code>remote path: %REMOTE%</code> <p>Either define local paths for these default variables or edit the script to provide actual paths.</p> <p>For example: local source, remote source</p> <p><i>localpath:c:\code samples\vea\php\sample</i></p> <p><i>remotepath:web server/sample</i></p> <p>where:</p> <ul style="list-style-type: none"> • <i>web server</i> is a network or local share • <i>sample</i> is a folder below share 	
PHP	<p>In order to debug PHP scripts in Enterprise Architect, it is a requirement that the PHP is configured properly to load the XDebug extension.</p> <p>Settings similar to those below should be used:</p> <ul style="list-style-type: none"> • <code>[xdebug]</code> • <code>xdebug.extended_info=1</code> • <code>xdebug.idekey=ea</code> • <code>xdebug.remote_enable=1</code> • <code>xdebug.remote_handler=dbgp</code> • <code>xdebug.remote_autostart=1</code> • <code>xdebug.remote_host=X.X.X.X</code> • <code>xdebug.remote_port=9000</code> • <code>xdebug.show_local_vars=1</code> <p>The IP address X.X.X.X refers to and should match the host specified in the model Analyzer Script.</p> <p>The IP address is the address XDebug connects with and the same address the Enterprise Architect PHP agent listens on.</p>	
Apache	<p>For debugging using Apache, the following lines should be present in the Apache configuration file, <i>httpd.conf</i>:</p> <pre>LoadModule php5_module "php_home/ php5apache2_2.dll" AddHandler application/x-httpd-php . php PHPLibrary "php_home"</pre>	

Topic	Detail	See also
	where <code>php_home</code> is the PHP installation path (the path where <code>php.ini</code> and <code>apache.dll</code> exist).	
Troubleshooting	To prevent both PHP and Apache timeouts during a debugging session, the settings below might require modification. These settings were used while developing the PHP Debugging agent in Enterprise Architect.	
	PHP File: php.ini <pre> ; EA prevent PHP timeouts when debugging PHP extensions max_execution_time = 0 ; EA prevent web server timeouts when debugging PHP extensions max_input_time = -1 ; EA log errors display_errors = On ; EA display startup errors display_startup_errors = On </pre>	
	Apache File: httpd.conf <pre> ; EA prevent timeouts while debugging php extensions Timeout 60000 </pre>	

Learning Center topics

- (Alt+F1) | **Build and Debug | Debug | PHP Samples**
- (Alt+F1) | **Build and Debug | Debug | PHP**

10.8.2.6.6 *The GNU Debugger*

Enterprise Architect supports debugging with the **GNU Debugger (GDB)**, which is a portable debugger that runs on Unix-like systems such as Linux, as well as Windows. This debugger works for many programming languages, including Ada, Java, C, C++ and Objective-C, and enables you to debug your applications locally or remotely.

This topic discusses the basic setup of the debugger.

Access **Analyzer | Execution Analyzer (Shift + F12) : New** (Toolbar icon) > **Debug > Platform Analyzer | Execution Analyzer (Shift + F12) : locate and double-click on required script > Debug > Platform**

Topics

Topic	Details	See also
Script Setup	An Analyzer Script is a basic requirement for debugging in Enterprise Architect; you create a script using the Execution Analyzer toolbar.	Managing Analyzer

Topic	Details	See also																					
	<p>On the Platform page of the Execution Analyzer Script Editor, in the Debugger field click on the drop-down arrow and select GDB.</p> <p>The property panel displays a number of connection settings, for which you provide values as indicated below:</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>path</td> <td><path></td> <td>The complete file path of the GDB executable; you use this if GDB cannot be found in the system path.</td> </tr> <tr> <td>remote</td> <td>F</td> <td>Set for remote debugging; otherwise leave blank.</td> </tr> <tr> <td>port</td> <td><nnnnn></td> <td>The port to connect to on the remote server.</td> </tr> <tr> <td>host</td> <td>localhost</td> <td>The hostname to connect to.</td> </tr> <tr> <td>fetch</td> <td>T</td> <td>Set to retrieve the binary from the remote system.</td> </tr> <tr> <td>dumpgdb</td> <td><path></td> <td>Filename to write the GDB output to.</td> </tr> </tbody> </table>	Field	Value	Meaning	path	<path>	The complete file path of the GDB executable; you use this if GDB cannot be found in the system path.	remote	F	Set for remote debugging; otherwise leave blank.	port	<nnnnn>	The port to connect to on the remote server.	host	localhost	The hostname to connect to.	fetch	T	Set to retrieve the binary from the remote system.	dumpgdb	<path>	Filename to write the GDB output to.	Scripts ^[1828]
Field	Value	Meaning																					
path	<path>	The complete file path of the GDB executable; you use this if GDB cannot be found in the system path.																					
remote	F	Set for remote debugging; otherwise leave blank.																					
port	<nnnnn>	The port to connect to on the remote server.																					
host	localhost	The hostname to connect to.																					
fetch	T	Set to retrieve the binary from the remote system.																					
dumpgdb	<path>	Filename to write the GDB output to.																					

10.8.2.6.7 Tracepoint Output

The Tracepoints page of the Analyzer Script enables you to direct where the output from any Trace statements goes during a debug session.

Access **Analyzer | Execution Analyzer (Shift + F12) : New** (Toolbar icon) > **Debug > Tracepoints**
Analyzer | Execution Analyzer (Shift + F12) : locate and double-click on required script > Debug > Tracepoints

Field	Detail	See also
Output	<p>You can select from two options:</p> <ul style="list-style-type: none"> • Screen (Default) - The output is directed to the Debug window • File - The output is directed to file 	
Folder	Enter the folder to use for Trace statement log files.	
Filename	Enter the name to use for the Trace statement log files.	

Field	Detail	See also
Overwrite	If selected, the specified file is overwritten each time a debug session is started.	
Auto Number	If selected, the Trace log file is composed of the filename you specify and a number. Each time you start a debug session, the number is incremented.	
Prefix trace output with function	If selected, any Trace statements executed during the debug session run are prefixed with the current function call.	

10.8.2.6.8 Workbench Setup

This topic describes the requirements for setting up the Object Workbench on Java and Microsoft .NET.

Access [Analyzer](#) | [Execution Analyzer](#) | [double-click on script](#) | [Debug](#) | [Workbench](#)

Topics

Topic	Detail	See also
Platforms Supported	The Workbench supports the following workbench platforms: <ul style="list-style-type: none"> Microsoft .NET (version 2.0 or later) excluding native C++, C and VB Java (JDK 1.4 or later) <p>The Execution Analysis script for the package must have a debugger configured.</p>	Debug ^[1837] (scripts)
Microsoft .NET Workbench	The .NET workbench requires an assembly, which is used to create the workbench items. You specify the path to the assembly on the Workbench page of the Analyzer Script. There are two constraints in using the .NET workbench: <ul style="list-style-type: none"> Members defined as <i>struct</i> in managed code are not supported Classes defined as <i>internal</i> are not supported 	General Setup for .NET ^[1850]
Java Workbench	The Java workbench uses the Virtual Machine settings configured in the Analyzer Script Debug page to create the JVM.	General Setup for Java ^[1843]

Learn more

- [How it Works](#) ^[2176]
- [Create & Delete Workbench Instances](#) ^[2177]

10.8.2.7 Add Run Command

This topic describes how to create a **command for running** your executable code.

Access **Analyzer | Execution Analyzer (Shift + F12) : New** (Toolbar icon) > **Run Analyzer | Execution Analyzer (Shift + F12) : locate and double-click on required script > Run**

Topics

Topic	Detail	See also
Usage	This is the command that is executed when you select the Analyzer Run menu option; at its simplest, the script would contain the location and name of the file to be run.	
Example	<p>The following two examples show scripts configured to run a .Net and a Java application in Enterprise Architect.</p> <p>.Net:</p> <pre>C: \ benchmark \ cpp \ example_net _1 \ release \ example.exe</pre> <p>Java:</p> <pre>customer</pre> <p>The command listed in this field is executed as if from the command prompt; as a result, if the executable path or any arguments contain spaces, they must be surrounded in quotes.</p>	

Notes

- Enterprise Architect provides the ability to start your application normally OR with debugging from the same script; the **Analyzer** menu has separate options for starting a normal run and a debug run

Learning Center topics

- (Alt+F1) | **Build and Debug | Overview | Run**

10.8.2.8 Add Deploy Command

This topic explains how to how to create a **command for deploying** the current package.

Access **Analyzer | Execution Analyzer (Shift + F12) : New** (Toolbar icon) > **Deploy**

Analyzer | Execution Analyzer (Shift + F12) : locate and double-click on required script > **Deploy**

Topics

Topic	Detail	See also
Usage	You create a script that is executed when you select the Analyzer Deploy menu option or when you press (Ctrl+Shift+Alt+F12)	
Execute Command as:	<p>Process</p> <p>If the deployment is handled externally, enter the path to the program or batch file to run followed by any parameters; the program is launched in a separate process.</p> <p>Example:</p> <pre>C:\apache-ant-1.7.1\bin\ant.cmd myproject deploy</pre> <p>Batch File</p> <p>When using this option, you can enter multiple commands that are then executed as a single script in a command console; you have access to any environment variables available in a standard command console.</p> <p>Example:</p> <pre>@echo on IF NOT EXIST "%1%" GOTO DEPLOY_NOWAR IF "%APACHE_HOME%" == "" GOTO DEPLOY_NOAPACHE xcopy /L "%1%" "%APACHE_HOME%\webapps" GOTO DEPLOY_END rem rem NO WAR FILE rem :DEPLOY_NOWAR echo "%1% WAR file not found" GOTO DEPLOY_END rem rem NO APACHE ENVIRONMENT VARIABLE rem :DEPLOY_NOAPACHE echo "APACHE_HOME environment variable not found" :DEPLOY_END pause</pre>	
Parse Output	<p>Selecting a Parser from the list causes output of the deploy script to be captured; the output is parsed according to the syntax selected from the list.</p> <p>The output window can be displayed by selecting View System Output.</p>	

10.8.2.9 Configure Recording Detail

To avoid generating too complex a Sequence diagram, you must control the scope of what is being recorded by:

- Filtering - using the Analyzer Script you can exclude specific modules, Classes and functions from the

recording

Access **Analyzer | Execution Analyzer (Shift + F12) Double-click on required Analyzer Script > Recording**

Topics

Topic	Detail	See also
Filtering	<p>If the Enable Filter checkbox is selected on the Recording page of the Execution Analyzer Script Editor, the debugger excludes calls to matching methods from the generated Sequence history and diagram; the comparison is case-sensitive.</p> <p>To add a value, click on the New (Insert) icon in the right corner of the Exclusion Filters box, and type in the comparison string; each filter string takes the form:</p> <pre>class_name_token : : method_name_token</pre> <p>The <i>class_name_token</i> excludes calls to all methods of a Class or Classes that have a name matching the token; the string can contain the wildcard character * (asterisk).</p> <p>The <i>method_name_token</i> excludes calls to methods having a name that matches the token; again, the string can contain the wildcard character * .</p> <p>Both tokens are optional; if no Class token is present, the filter is applied only to global or public functions (that is, methods not belonging to any Class).</p>	<p>Managing Analyzer Scripts ^[1828]</p> <p>Analyzer Script Editor ^[1831]</p>
Example	<p>In the Java example below, the debugger would exclude:</p> <ul style="list-style-type: none"> • Calls to the <i>OnDraw</i> method for the Class <i>Example.common.draw.DrawPane</i> • Calls to any method of any Class having a name beginning with <i>Example.source.Collection</i> • Calls to any constructor for any Class (such as <i><clint></i> and <i><init></i>) <div data-bbox="422 1624 874 1780" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p>Filters</p> <p>Example.common.draw.DrawPane::OnDraw</p> <p>Example.source.Collection*</p> <p>*::init*</p> </div> <p>In the Native Code example below, the debugger would exclude:</p> <ul style="list-style-type: none"> • Calls made to Standard Template Library namespace • Calls to any Class beginning with <i>TOb</i> • Calls to any method of Class <i>CLock</i> • Calls to any Global or Public Function with a name beginning 	

Topic	Detail	See also
	<p>with <i>Get</i></p> <ul style="list-style-type: none"> • Calls to the method <i>GetLocation</i> for Class <i>Ctrain</i> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p>Filters</p> <p>std*</p> <p>TOb*</p> <p>CLock</p> <p>CTrain::GetLocation</p> <p>::Get*</p> </div>	

Reference

To Filter	Use Filter Entry
All public functions having a name beginning with Get from the recording session (for example, <i>GetClientRect</i> in Windows API).	:: Get *
All methods beginning with Get for every Class member method.	* :: Get *
All methods beginning with Get from the Class <i>CClass</i> .	CClass :: Get *
All methods for Class <i>CClass</i> .	CClass :: *
All methods for Classes belonging to Standard Template and Active Template Libraries.	ATL* std*
The specific method <i>GetName</i> for Class <i>CClass</i> .	CClass :: Get Name



Learn more

- [Control Stack Depth](#)  2153

10.8.3 Build Application

This topic explains how to execute a Build script on your application, within Enterprise Architect.

Access **Analyzer | Execution Analyzer (Shift+F12) : select required script, then:**

- 
- **(Ctrl + Shift + F12)**
- **Analyzer | Build**
- Right-click and select either **Build** or **Rebuild** context menu option, or
- **Analyzer | Debug | Debugger:**  > **Build**

Guide:

When you select the **Build** option, it executes the Build command in the script selected in the Execution Analyzer window. The progress and outcome of the build operation are displayed in the Build tab of the Output window.

Locate any compilation errors and correct the code, and repeat the build-correct cycle until the compilation log on the Output window Build tab shows no compile errors.

Learn more

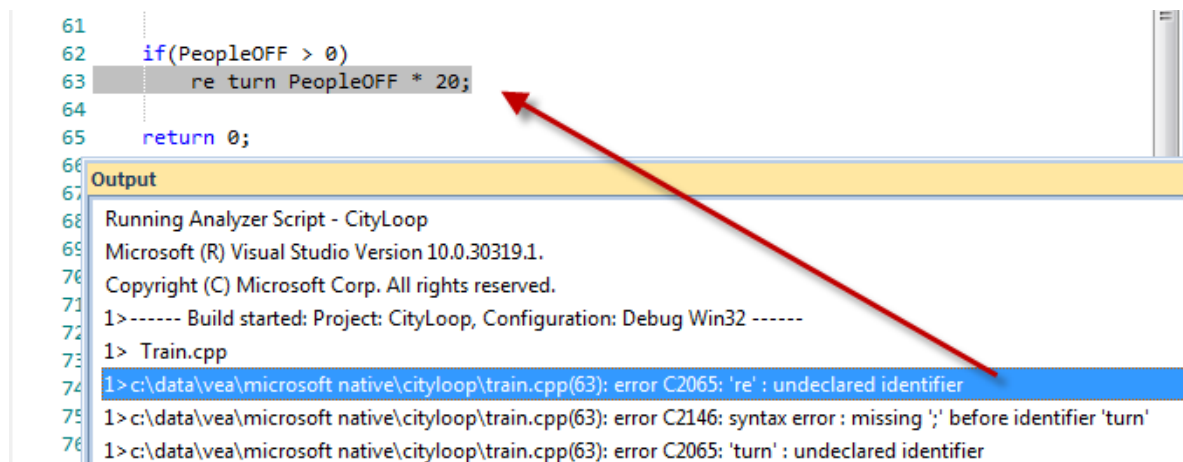
- [Add Build Commands](#) ¹⁸³²
- [Analyzer Script Editor](#) ¹⁸³⁷
- [Locate Compiler Errors in Code](#) ¹⁸⁶⁶

10.8.3.1 Locate Compiler Errors in Code

When you build an application using a Build script, the compiler output is logged in Enterprise Architect's Output window. The output includes any compilation error messages that were generated.

Access **View | System Output**

If you double-click on a compiler error message, Enterprise Architect loads the appropriate source file and positions the cursor on the line in the code at which the error was reported, as illustrated below:



```
61
62     if(PeopleOFF > 0)
63         re turn PeopleOFF * 20;
64
65     return 0;
66
67 Output
68 Running Analyzer Script - CityLoop
69 Microsoft (R) Visual Studio Version 10.0.30319.1.
70 Copyright (C) Microsoft Corp. All rights reserved.
71 1>----- Build started: Project: CityLoop, Configuration: Debug Win32 -----
72
73 1> Train.cpp
74 1> c:\data\vea\microsoft native\cityloop\train.cpp(63): error C2065: 're' : undeclared identifier
75 1> c:\data\vea\microsoft native\cityloop\train.cpp(63): error C2146: syntax error : missing ';' before identifier 'turn'
76 1> c:\data\vea\microsoft native\cityloop\train.cpp(63): error C2065: 'turn' : undeclared identifier
```

Learn more

- [Build Application](#)^[1866]
- [The Output Window](#)^[164]

10.8.4 Debugging

This section describes how you define and execute the debugging actions:

- [Before Starting](#)^[1867]
- [Debug](#)^[1837] (scripts)
- [Breakpoint and Marker Management](#)^[1869]
- [Debugger Facilities](#)^[1874]
- [Run a Debug Session](#)^[1890]

Notes

- You can register the Enterprise Architect debugger as the operating system just-in-time debugger, to be invoked when an application running outside Enterprise Architect on the system either encounters an exception or crashes

Learn more

- [Debug Menu](#)^[122]

10.8.4.1 Before Starting

The Model Driven Development Environment provides Debuggers for the following frameworks:

- Microsoft Native Code applications
- Microsoft .NET applications
- Java applications
- PHP applications

Checklist:

Step	Action	See also
1	A model must be open.	
2	Ensure any source code for the areas of interest has been generated or imported into the model, synchronized with the model and, if necessary, compiled.	Generate Source Code <small>[1768]</small> Importing Source Code <small>[1790]</small> Update Package Contents <small>[1774]</small>

Option 1 - No Analyzer Script:

If you have *not* configured an Analyzer Script, you can still debug a process that is running.

Step	Action	See also
1	Optionally set breakpoints before debugging.	Breakpoint and Marker Management <small>[1869]</small>
2	Use the Attach button on the Debug toolbar <ul style="list-style-type: none"> Select the debugging platform Select the process to debug 	

Option 2 - Analyzer Script, with Debug Page Configured:

Step	Action	See also
1	Ensure the application has been built. You can do this internally using a Build Script, or you can build the application externally; the important thing is that the application is built on the latest versions of the source.	Build Application <small>[1866]</small> Add Build Commands <small>[1832]</small> (script)
2	Create an appropriate debug script on the Debug page of the Analyzer Script Editor.	Debug <small>[1837]</small> (script)
3	Set breakpoints in the code.	Set Code Breakpoints <small>[1871]</small>
4	Start the debugger.	Run the Debugger <small>[1875]</small>


10.8.4.2 Breakpoint and Marker Management




Breakpoints work in Enterprise Architect much like in any other debugger. Adding a breakpoint notifies the debugger to trap code execution at the point you have specified. When a breakpoint is encountered by a thread of the application being debugged, the source code is displayed in the Source Code Editor, and the line of code where the breakpoint occurred is highlighted.

Selecting a different package in the project affects which breakpoints are displayed. An Enterprise Architect model maintains breakpoints for every package having a *Build Script - Debug* command.





Access [Analyzer | Breakpoints & Events](#)

Breakpoint and Markers Options:

Option	Detail	See also
Delete a breakpoint or marker	<p>To delete a specific breakpoint:</p> <ul style="list-style-type: none"> • If the breakpoint is enabled, click on the red breakpoint circle in the left margin of the Source Code Editor, or • Right-click on the breakpoint or marker in the Source Code Editor, the <i>Breakpoints</i> folder or the Breakpoints & Events window and select the Delete context menu option, or • Select the breakpoint in the Debug Breakpoints tab and press (Delete) 	Setting Code Breakpoints ^[187]
Delete all breakpoints	Click on the Delete all breakpoints button on the Breakpoints & Events window toolbar ().	
Convert breakpoint to Start Recording marker or End Recording marker	<p>(For Visual Execution Analysis, when recording execution to generate a Sequence diagram).</p> <p>In the <i>Breakpoints</i> folder or the Breakpoints & Events window, right-click on the breakpoint and select the context menu option to convert it to either a Start Recording marker or an End Recording marker.</p>	Recording Sequence Diagrams ^[214]
Disable a breakpoint	<p>Deselect the checkbox against the breakpoint or marker, on the Breakpoints & Events window.</p> <p>The breakpoint is then shown as an empty grey circle.</p>	
Enable a breakpoint or marker	Select the checkbox against the breakpoint or marker, on the Breakpoints & Events window.	
Disable all	Click on the Disable all breakpoints button in the Breakpoints &	

Option	Detail	See also
breakpoints	Events window toolbar ()	
Enable all breakpoints	Click on the Enable all breakpoints button in the Breakpoints & Events window toolbar ()	
Change the stack depth	Set the number of stack levels to record to, relative to the level at which recording starts.	Control Stack Depth ^[2153]
Identify or change the marker set	<p>Check the <input type="text" value="Default"/> field in the Breakpoints & Events window toolbar.</p> <p>If necessary, click on the drop down arrow and select a different marker set.</p> <p>The Default set is normally used for debugging and is personal to your user ID; other marker sets are shared between all users within the model.</p>	Working with Marker Sets ^[2158]
Change how breakpoints and markers are grouped on the Breakpoints & Events window	<p>The breakpoints and markers can be grouped by Class or by code file.</p> <p>To group the items, click on the down arrow on the  icon in the toolbar, and click on the appropriate option.</p> <p>If you do not want to group the items, click on the selected option to deselect it; the breakpoints and markers are line listed by line number.</p>	

Breakpoint States:

DEBUGGER STATE			
	Running	Not running	See also
	Bound.	Enabled.	
	Disabled.	Disabled.	
	Not bound - this usually means that the DLL is not yet loaded or was not built with debug information.	N/a	
	Failed - this usually means a break could not be set at this time.	N/a	Failure to Bind Breakpoint ^[1873]

[Learn more](#)

- [Marker Types](#) ^[2155]

10.8.4.2.1 Setting Code Breakpoints

Normal Breakpoints are typically set on a line of source code. When the Debugger hits the indicated line during normal execution, the Debugger halts execution and displays the local variables, call stack, threads and other run-time information.

How to

To set a breakpoint on a line of code

Step	Action	See also
1	Open the source code to debug in the integrated source code editor.	
2	<p>Find the appropriate code line and click in the left margin column - a solid red circle in the margin indicates that a breakpoint has been set at that position.</p> <p>If the code is currently halted at a breakpoint, that point is indicated by a blue arrow next to the marker.</p> <p>Alternatively, you can set the Breakpoint marker (or other marker) by right-clicking on the left margin on the required line, to display the breakpoint/marker context menu; select the appropriate marker type.</p>	<p>Set Record Markers ^[2155]</p>

Learning Center topics

- (Alt+F1) | **Build and Debug** | **Debug** | **Add Breakpoint**

10.8.4.2.2 Breakpoint Properties

Breakpoints in Enterprise Architect support a number of additional properties that change when a breakpoint is hit, and determine what happens when it is hit.

Access **Breakpoint Context Menu** | **Properties**

Use to

- Change breakpoint actions

- Move the breakpoint to another line
- Add constraints to restrict when a breakpoint is hit
- Add information to be logged when the breakpoint is hit

Reference

Field	Usage	See also
Action	Change the behavior when the breakpoint is hit.	Marker Types ^[2155]
Line	Change the line that this breakpoint is on.	
Constraints	Associate a condition with the breakpoint. The program is allowed to run if the condition is not met when the breakpoint is encountered.	
Log	Log information to the Debug window when the breakpoint is hit.	Run the Debugger ^[1875]

Specifying constraints:

You can compose constraints using any variables that are in scope at the time the breakpoint occurs.

```
m_strName="John"
```

```
m_strName=""
```

```
m_nCount > 0
```

You can also join constraints using logical operands, for example:

```
m_strName="John" AND m_nCount > 0
```

You can access elements of an array or pointer, for example:

```
m_pNames[10]="John"
```

You can also nest constraints using parentheses to impose precedence, for example:

```
( x>0 AND y>0) OR (x < 0 AND y < 0)
```

Adding log information:

Add your trace statement into the Log field. To output a string variable, prefix the variable name with the '\$' token and for primitive types use an '@' token .

Example for outputting a string member named `m_Name` of an object named `X`.

```
The value of m_Name = $X.m_Name
```

Example for outputting an integer named `m_Count` of an object named `X`.

The value of `m_Count` = `@X.m_Count`

Learning Center topics

- (Alt+F1) | [Build and Debug | Debug | Breakpoint Properties](#)

10.8.4.2.3 Failure to Bind Breakpoint

Topics

Topic	Detail	See also
Breakpoint Failures	<p>A breakpoint failure occurs if there is a problem in binding the breakpoint.</p> <p>A warning message displays in the Details column of the Breakpoints & Events window, identifying the type of problem</p> <ul style="list-style-type: none"> • The source file for the breakpoint does not match the source file used to build the application image • The time date stamp on the file is greater than that of the image <p>A warning message is also output to the Debug window.</p>	<p>Run the Debugger <small>[1873]</small></p>

10.8.4.2.4 Trace Statements

Tracepoints are set in the code editor. Like breakpoints, they are placed on a line of code. When that line of code executes, the Execution Analyzer evaluates and prints the statement either to the Debug window or to file, if this has been configured in the Analyzer script.

How to

To set Tracepoints

Step	Action	See also
1	Open the source code to debug in the source code editor.	Editing Source Code <small>[1799]</small>
2	Find the appropriate code line, right-click in the left margin and select the Add Tracepoint Marker option.	

Step	Action	See also
3	<pre> 55 DWORD CTrain::Disembark(int PeopleOFF) 56 { 57 if(Passengers - PeopleOFF > -1) 58 Passengers -= PeopleOFF; 59 else 60 Passengers = 0; 61 62 if(PeopleOFF > 0) 63 return PeopleOFF * 20; 64 65 return 0; 66 } </pre>	
4	Right-click on the Tracepoint and choose the Properties context menu option.	
5	In the Log field type the required Trace statement.	Testpoint Management <small>[2188]</small>

Specifying Tracepoints:

Using the example in the image above, we could output the number of people getting off a train by using the following statement:

There were @Passengers before @PeopleOFF got off the train

To print a variable, precede the variable name with the @ token. If a variable represents a string, precede the variable with a \$ token.

Learning Center topics

- (Alt+F1) | **Build and Debug** | **Debug** | **Add Tracepoint**

10.8.4.3 Debugger Facilities

This section describes the facilities available while running a debug session. It covers:

- [Run the Debugger](#)^[1875]
- [View the Local Variables](#)^[1877]
- [View the Call Stack](#)^[1884] (and [Create Sequence Diagram of Call Stack](#))^[1885]
- [Inspect Process Memory](#)^[1886]
- [Show Loaded Modules](#)^[1887]
- [Debug Another Process](#)^[1888]
- [Process First Chance Exceptions](#)^[1889]

Learn more

- For an outline of how you might use the facilities together or in sequence during a debug session, see [Run a Debug Session](#)^[1890]





10.8.4.3.1 Run the Debugger





This topic describes how to start and control a debug session.


(Note that many of these facilities are also available through the Debugger Toolbar (**View | Toolbars | Debug & Record**))

Access **Analyzer | Debug | Debugger**
Analyzer Function keys

Using the Debug window

Topic	Detail	See also
Start the Debugger	<p>If you have configured an Analyzer script for your target application and you have made it the default script, you can start debugging the application immediately in the following ways:</p> <ul style="list-style-type: none"> Click on the  button Press (F6), or Select the Analyzer Debug Run menu option <p>If you have not made your target application the default script it is possible to start any Execution Analysis script by right clicking on it in the Execution Analysis window and selecting the "Debug" option.</p> <p>If you have no Analyzer Script, it is still possible to debug a running application by attaching to that process directly:</p> <ul style="list-style-type: none"> Click on the  (Attach) button and choose the debugging platform to use 	<p>Before Starting^[1867]</p> <p>Analyzer Script Editor^[1831]</p> <p>Debug Another Process^[1888]</p>
Pause/Resume Debugging	To pause a debugging session, or to resume the session after pausing, click on the  button.	
Stop the Debugger	<p>To stop debugging, either:</p> <ul style="list-style-type: none"> Click on the  (Stop) button Press (Ctrl+Alt+F6), or Select the Analyzer Debug Debug Stop menu option <p>The debugger normally ends when the current debug process terminates, however some applications and services (e.g. Java Virtual Machine) may require the debugger to be manually Stopped.</p>	<p>Breakpoint and Marker Management^[1869]</p>

Topic	Detail	See also
Step Over Lines of Code	<p>To step over the next line of code:</p> <ul style="list-style-type: none"> • Click on the  (Step Over) button • Press (Alt+F6), or • Select the Analyzer Debug Step Over menu option 	
Step Into Function Calls	<p>To step into a function call:</p> <ul style="list-style-type: none"> • Click on the  (Step In) button • Press (Shift+F6), or • Select the Analyzer Debug Step In menu option <p>If no source is available for the target function then the debugger returns immediately to the caller.</p>	
Step Out Of Functions	<p>To step out of a function:</p> <ul style="list-style-type: none"> • Click on the  (Step Out) button • Press (Ctrl+F6), or • Select the Analyzer Debug Step Out menu option <p>If the debugger steps out into a function with no source code, it will continue to step out until a point is found that has source code.</p>	
Show Execution Point	<p>To return to the source file and line of code that the debugger is currently executing, click on the  (Show Execution Point) button.</p> <p>The appropriate line is highlighted, with a pink arrow in the left margin of the screen.</p>	
Output	<p>During a debug session, messages display in the Debug window detailing:</p> <ul style="list-style-type: none"> • Startup of session • Termination of session • Exceptions • Errors • Trace messages, such as those output using <i>Java System.out</i> or <i>.NET System.Diagnostics.Debug</i> 	

Topic	Detail	See also
	<p>If you double-click on a debug message, either:</p> <ul style="list-style-type: none"> • A pop-up displays with more complete message text, or • If there has been a memory leak, the file is displayed at the point at which the error occurred 	
Save Output (and Clear Output)	<p>You can save the entire contents of the Debug output to an external .txt file, or you can save selected lines from the output to the Enterprise Architect clipboard.</p> <p>To save all of the output to file, click on the  (Save output to file) button.</p> <p>To save selected lines to the clipboard, right-click on the selection and select the Copy Selected to Clipboard context menu option.</p> <p>When you have saved the output or otherwise do not want to display it any more, right-click on the current output and select the Clear Results context menu option.</p>	

Learn more

- [Debug & Record Toolbar](#)^[150]
- [View the Local Variables](#)^[1877]
- [View the Call Stack](#)^[1884]

10.8.4.3.2 View the Local Variables

Whenever a thread encounters a breakpoint, the Locals window displays all the local variables for the thread at its current stack frame.

When a **Simulation** is active, the Local Variables displayed will be the current variables defined within the Simulation context using Javascript.

Access **Analyzer | Locals**

Analyzer | Execution Analyzer:  **| Locals**

Topics

Topic	Detail	See also				
Display Icons	<p>The value and type of any in-scope variables are displayed in a tree; each variable has a colored box icon that identifies the type of variable:</p> <table border="1" data-bbox="475 1888 919 2029"> <thead> <tr> <th>Color</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>Blue</td> <td>Object with members</td> </tr> </tbody> </table>	Color	Meaning	Blue	Object with members	
Color	Meaning					
Blue	Object with members					

Topic	Detail		See also												
	<table border="1"> <thead> <tr> <th>Color</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> <tr> <td>Green</td> <td>Arrays</td> </tr> <tr> <td>Pink</td> <td>Elemental types</td> </tr> <tr> <td>Yellow</td> <td>Parameters</td> </tr> <tr> <td>Red</td> <td>Workbench Instance</td> </tr> </tbody> </table>	Color	Meaning			Green	Arrays	Pink	Elemental types	Yellow	Parameters	Red	Workbench Instance		
Color	Meaning														
Green	Arrays														
Pink	Elemental types														
Yellow	Parameters														
Red	Workbench Instance														
Item Context Menu	Option	Detail													
	Break When Variable is Modified	Set data breakpoints on the selected memory variable, to halt debugger execution at the line of code that has just caused the value of the variable to change.	Break When a Variable Changes Value ^[1882]												
	View Memory at Address	Display the raw values in memory at the selected address, in hex and ASCII.	Inspect Process Memory ^[1886]												
	View in Editor	Display the variable string in the String Viewer dialog.	View Content Of Long Strings ^[1879]												
	Dump Variable Members to File	Capture and store the selected variables to a separate location; a browser displays to select the appropriate .txt file name and file path.													
	Save Snapshot of Variable	Capture the value of a variable at a specific point in the life of that variable.	Variable Snapshots ^[1881]												
	Compare Variable Snapshots	Compare the values of a variable at different points in the life of that variable.													
	Copy	Copy the selected variable to the Enterprise Architect clipboard.													

Learn more

- [View Debug Variables in Code](#)^[1879]

- [View Variables in Other Scopes](#)^[1880]
- [Breakpoint and Marker Management](#)^[1869]
- [View the Call Stack](#)^[1884]

Learning Center topics

- (Alt+F1) | **Build and Debug** | **Debug**

10.8.4.3.2.1 View Content Of Long Strings

For efficiency, the Locals window only shows partial strings. However, the size of any variable value displayed in the window can be up to 256 characters.

You can display the entire content of long strings on the String Viewer dialog.

Access **Analyzer** | **Locals: right-click on variable** | **View in Editor**

Analyzer | **Execution Analyzer:**  | **Locals: right-click on variable** | **View in Editor**

10.8.4.3.2.2 View Debug Variables in Code Editors

Topics

Topic	Detail	See also
Inspecting Member Variables	<p>During debugging, whenever a thread is suspended at a line of execution, you can inspect member variables in the Source Code Editor.</p> <p>To evaluate a member variable, use the mouse to move the cursor over the variable in the Editor, as shown in the following examples:</p> <pre> public void Print() { int n = 0; while (names[n].Length > 0) { names = {[4] names[0]=book, names[0]=book, names[1]=novel, names[2]=film, ...}; Document d = new Document (names[n++]); d.Print (); } } public void Print() { int n = 0; while (32-bit signed integer n=0 0) { Document d = new Document (names [n++]); d.Print (); } } </pre>	<p>View the Local Variables^[1877]</p> <p>View Variables in Other Scopes^[1880]</p>

10.8.4.3.2.3 View Variables in Other Scopes

Access **Analyzer | Watches**

Analyzer | Execution Analyzer:  **| Watches**

Topics

Topic	Detail	See also
Watches	<p>The Watches window is most useful for native code (C, C++, VB) where it can be used to evaluate data items that are not available as Local Variables - data items with module or file scope and static Class member items.</p> <p>You can also use the window to evaluate static Class member items in Java and .NET</p> <p>To use the Watches window, type the name of the variable to examine in the field in the window toolbar, and either press (Enter) or click on the Add new watched item icon.</p> <p>To examine a static Class member variable in C++, Java or Microsoft .NET, enter its fully qualified name</p> <pre>CMyCl ass : : MySt at i cVar</pre> <p>To examine a C++ data symbol with module or file scope, just enter its name.</p> <p>Variables are evaluated by looking at the current scope; that is, the module of the current stack frame (you can change the scope at a breakpoint by double-clicking the frame in the Call Stack).</p> <p>If the global variable exists in a different module, you can examine the variable by prefixing the module name to the variable</p> <pre>modul ename! var i abl e_ name</pre> <p>The names of the items to evaluate persist for the package and user ID, so the next time <i>you</i> debug the same project, the items evaluate automatically whenever a breakpoint occurs; they do not appear if another user debugs the same code.</p> <p>If necessary, you can delete items using the Delete all watched items icon in the toolbar, or the Delete Watch (for a selected watched item) or Delete All Watches context menu options inside the window.</p> <p>For other context menu options, see the <i>View the Local Variables</i> topic.</p>	<p>View Elements of Array^[1881]</p> <p>View the Local Variables^[1877]</p> <p>View Debug Variables in Code Editors^[1879]</p> <p>Inspect Process Memory^[1886]</p>

Learning Center topics

- (Alt+F1) | **Build and Debug | Debug | View Global Variable**

Access Analyzer | Watches

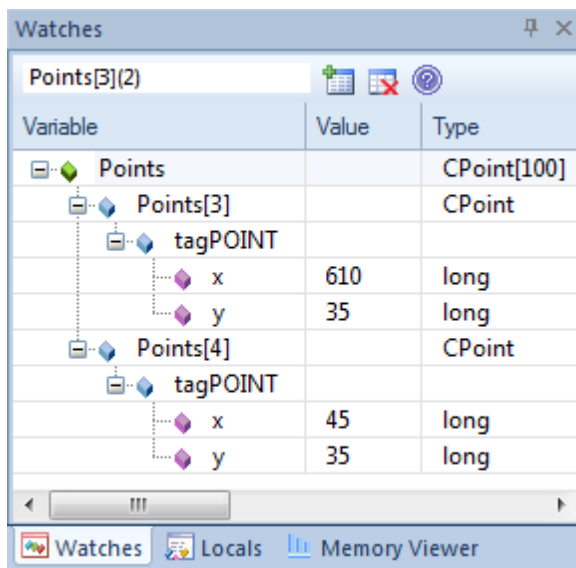
You can use the Watches window to inspect one or more specific elements of an array.

In the field to the left of the Watches window toolbar, type the name of the variable (the array) followed by the start element and, optionally, the number of elements to display. The start element is enclosed in square brackets and the count of elements is enclosed in parentheses; that is:

```
variable[ start_element ]( count_of_elements )
```

For example:

Points[3](2) displays the third and fourth array elements of the Points array, as illustrated below.



If you entered Points[3] the Watches window would show the third array element only.

Learn more

- [View Variables in Other Scopes](#) ^[1880]
- [View the Local Variables](#) ^[1877]
- [View Debug Variables in Code Editors](#) ^[1879]

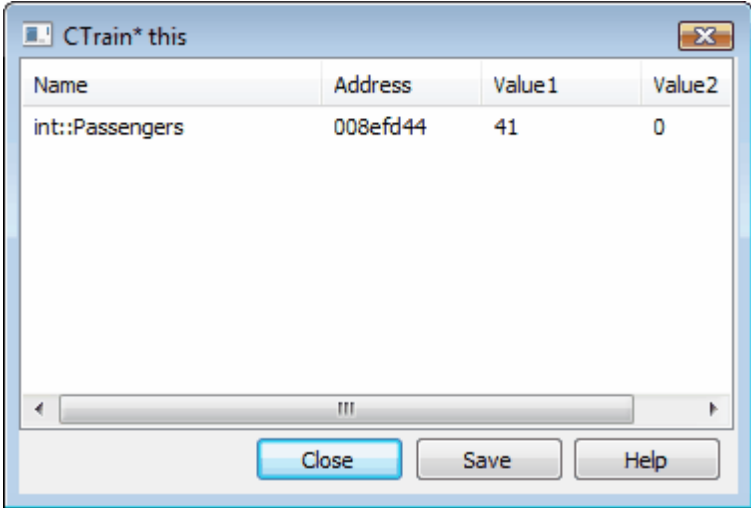
10.8.4.3.2.4 Variable Snapshots

Snapshots enable you to compare the value of a variable at different points in the life of that variable. You can take a snapshot of a variable when the program is at a breakpoint.

The debugger does not copy the value of the selected variable only; for complex variables it copies the values of the selected variable and of each of its hierarchy of members until it can find no debug information for a member or no more members can be found.

How to

To capture variable snapshots

Step	Action	See also
1	In the Code Editor, set two breakpoints: start at the start of a function and end at the end of the function.	Setting Code Breakpoints ^[187]
2	At the start breakpoint, right-click on a variable in the Locals window and select the Save Variable Snapshot context menu option.	View the Local Variables ^[187]
3	Run the application.	
4	<p>When the end breakpoint is reached, right-click on the variable in the Locals window and select the Compare Variable Snapshots context menu option.</p> <p>A dialog displays that shows the original value from the first snapshot and the current value from the second snapshot, as illustrated below (this example is taken from the <i>EA.Example</i> model).</p> 	

Notes

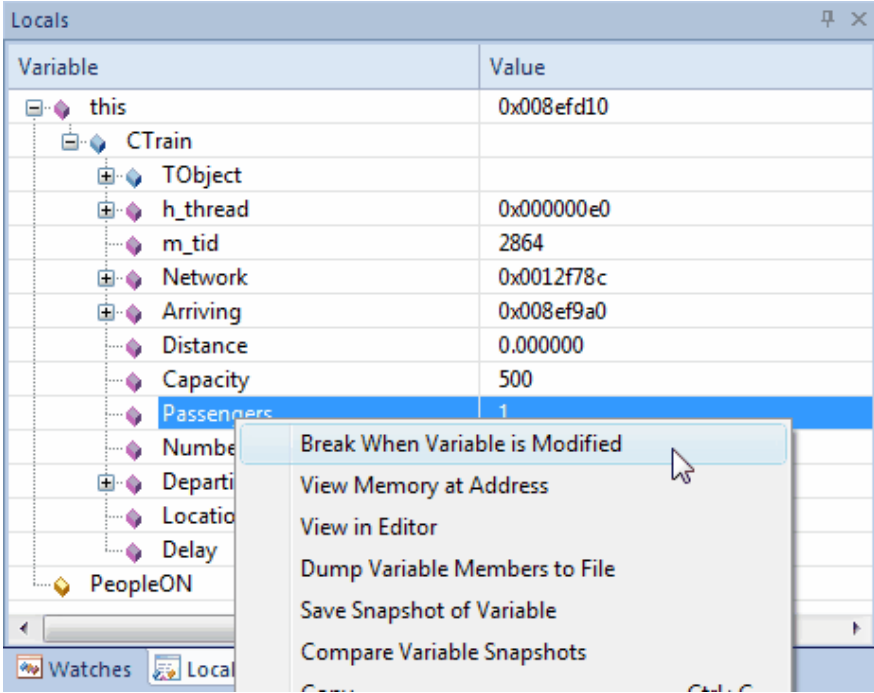
- This facility compares the values of a variable by address; be aware that if there are different threads with different instances of the same variable name, these are not comparable

10.8.4.3.2.5 Break When a Variable Changes Value

Data breakpoints can be set on a pre-determined memory variable to cause the debugger to halt execution at the line of code that has just caused the value of the variable to change. This can be useful when trying to track down the point at which a variable is modified during program execution, especially if it is not clear how program execution is affecting a particular object state.

Access **Analyzer | Locals: right-click on variable | Break When Variable is Modified**
Analyzer | Watches: right-click on variable | Break When Variable is Modified

Topics

Topic	Detail	See also
Usage	<p>In order to establish a data breakpoint you must first set a normal breakpoint at a point in the code that presents the required scope of local variables to choose from.</p> <p>Execute the debugger (press F6); once the program has halted execution, right-click on the variable of interest and select the Break When Variable is Modified context menu option.</p> <p>There is no breakpoint indicator in the code, but in the Breakpoints & Events window, the data breakpoint displays as a blue circle in the Enabled column.</p> <p>The example below creates a notification on a variable from the Locals window; the item being watched is the <i>Passengers</i> integer member of the <i>CTrain</i> Class.</p>  <p>Continue to execute the debugger; when the variable changes and the debugger halts, the line of code is highlighted in the Source Code Editor and the log in the Debug window prints a statement reporting the effect.</p>	<p>Setting Code Breakpoints ^[1871]</p> <p>View the Local Variables ^[1877]</p> <p>View Variables in Other Scopes ^[1880]</p>

Notes

- This feature is not presently supported by the Microsoft .NET platform

Learning Center topics

- [\(Alt+F1\) | Build and Debug | Debug | Add Data Breakpoint](#)

10.8.4.3.3 View the Call Stack

The Call Stack window is used to display all currently running threads in a process. It can be used to identify which thread is operational immediately before program failure occurs.

When a **Simulation** is active, the Call Stack will show the current execution context for the running simulation. This will include a separate context stack for each concurrent simulation "thread".

A stack trace is displayed whenever a thread is suspended, through one of the step actions or through encountering a breakpoint. The Call Stack window can record a history of stack changes, and enables you to generate Sequence diagrams from this history.





Access [Analyzer | Call Stack](#)

Use to

- View stack history to understand the execution of a process
- View threads
- Save a call stack for later use
- Record call stack changes for Sequence diagram generation
- Generate a Sequence diagram from the call stack
- View the related code line in the Source Code Editor

Topics

Topic	Detail	See also
Indicators	<ul style="list-style-type: none"> • A pink arrow highlights the current stack frame • A blue arrow indicates a thread that is running • A red arrow indicates a thread for which a stack trace history is being recorded 	
Save a Call Stack to a .TXT File	<i>Not currently available.</i>	
Record a Thread in a Debug Session	<p>To record the execution of a <i>thread</i> and direct the recording to the Record & Analyze window, right-click on the thread in the call stack and select the appropriate context menu option:</p> <ul style="list-style-type: none"> • Record - to manually record the current thread during the debug session <p>Used in conjunction with the 'step' buttons of the debugger; each function that is called due to a step command is logged to the Record & Analyze window.</p> <ul style="list-style-type: none"> • Auto-Record - to perform auto-recording during a debug 	The Recording History ^[2149]

Topic	Detail	See also
	<p>session</p> <p>When you select this icon, the Analyzer begins recording and does not stop until either the program ends, you stop the debugger or you click on the Stop icon.</p>	
Stop Recording	<p>If you have started a manual or automatic recording of a thread you can stop it before completion; select the thread (indicated by a red arrow) and either:</p> <ul style="list-style-type: none"> • Click on the  (Stop Recording) button in the toolbar or • Right-click and select the Stop context menu option 	
Generate a Sequence Diagram from the Call Stack	<p>To generate Sequence diagram from the call stack trace, either:</p> <ul style="list-style-type: none"> • Click on the  (Generate Sequence Diagram of Stack) button, or • Right-click and select the Generate Sequence Diagram context menu option 	Create Sequence Diagram of Call Stack ^[1886]
Copy Stack to Recording History	<p>To add the stack details immediately to the Record & Analyze window (for later generation of Sequence diagrams) either:</p> <ul style="list-style-type: none"> • Click on the  (Copy Stack to Recording History) button, or • Right-click and select the Copy Stack to Record History context menu option 	The Recording History ^[2149]
Toggle Stack Depth	<p>To toggle between showing the full stack and showing only frames with source, click on the  (Toggle Stack Depth) button.</p>	
Display Related Code in Source Code Editor	<p>Double-click on a thread/frame to display the related line of code in the Source Code Editor; local variables are also refreshed for the selected frame.</p>	Editing Source Code ^[1799]

Learn more

- [Breakpoint and Marker Management](#)^[1869]

10.8.4.3.3.1 Inspect Process Memory

Using the Memory Viewer, you can display the raw values of memory in hex and ASCII. You can manually define the memory address in the **Address** field (top right), or right-click on a variable in the Locals or Watches window and select the **View Memory at Address** context menu option.

Access **Analyzer | More Analyzer Windows | Memory Viewer**
Right-click on the item | View Memory at Address

Notes

- The Memory Viewer is available for debugging Microsoft Native Code Applications (C, C++, VB) running on Windows or within WINE on Linux

Learn more

- [View the Local Variables](#) ^[1877]
- [View Variables in Other Scopes](#) ^[1880]

Learning Center topics

- (Alt+F1) | **Build and Debug | Debug | View Memory**

10.8.4.3.3.2 Create Sequence Diagram of Call Stack


The Call Stack window records a history of stack changes from which you can generate Sequence diagrams.

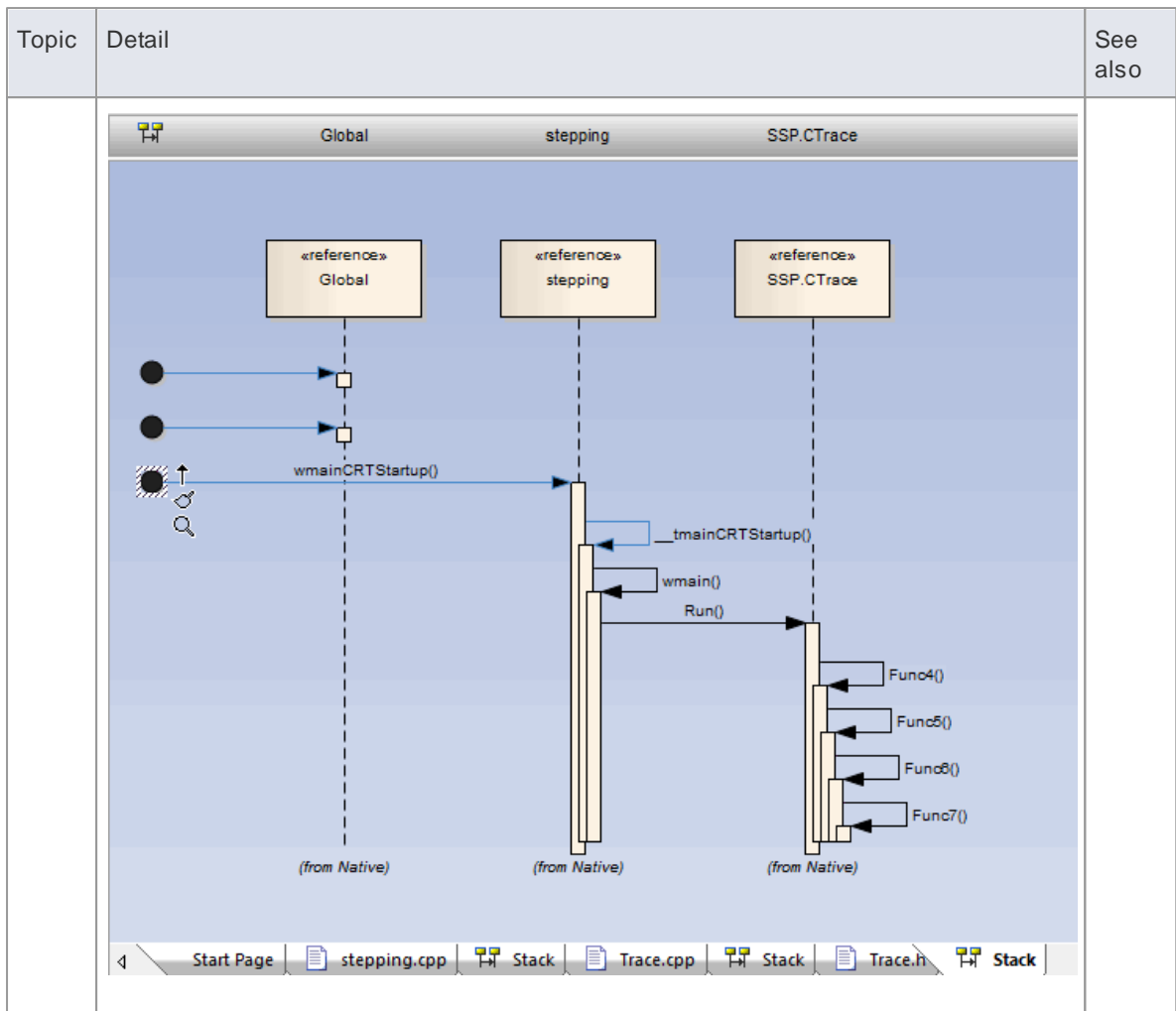
Access **Analyzer | Call Stack**

Use to

- Record Call Stack changes for Sequence Diagram generation
- Generate a Sequence Diagram from the Call Stack

Topics

Topic	Detail	See also
Usage	To generate a Sequence diagram from the current Stack, click on the  (Generate Sequence Diagram of Stack) button on the Call Stack window toolbar. This immediately generates a Sequence diagram in the Diagram View.	



Learn more

- [View the Call Stack](#)¹⁸⁸⁴

10.8.4.3.4 Show Loaded Modules

For .NET and native Windows applications, the Modules window lists those DLL's loaded by the debugged process. Associated symbolic files (PDB files) used by the debugger may also show up here.

Access [Analyzer](#) | [More Analyzer Windows](#) | [Modules](#)

Reference

The columns on the Modules window are described below:

Column	Action
Path	Determine the file path of the loaded module.
Modified Date	Determine the local file date and time the module was modified.

Column	Action
Debug Symbols	Establish the debug symbols type, whether debug information is present in the module and whether line information is present for the module (required for debugging).
Symbol File Match	Check the validity of the symbol file; if the value is false, the symbol file is out of date.
Symbol Path	Determine the file path of the symbol file, which must be present for debugging to work.
Modified Date	Determine the local file date and time the symbol file was created.

Learn more

- [Run a Debug Session](#) 

10.8.4.3.5 Debug Another Process

Rather than starting a process explicitly from within Enterprise Architect, you might want to debug an application (process) that is already running on your system.


In this case you can use the debugging capability to 'attach' to the process that is already running. Provided you have the appropriate debug information written into the running process, and/or associated debug files (such as .PDB files), the debugger binds to that process and initiates a debug session.

You can also 'detach' from the process after you have completed your inspection and leave the process to run as normal.

Access [Analyzer | Debug | Debugger > Debug window toolbar:](#) 

Topics

Topic	Detail	See also
Show Processes	When you select to debug another process, the Attach To Process dialog displays. You can limit the processes displayed using the radio buttons at the top of the dialog; to find a service such as Apache Tomcat or ASP.NET, select the System radio button.	

Topic	Detail	See also
Select Debugger	When you select a process, you may need to choose the debugger from the Debugger dropdown list; however, if the selected package has already been configured in an Analyzer Script, then the debugger listed in the script is preset on the dialog.	
Process Selection	Once you double-click on a process containing debug information, and Enterprise Architect is attached to the process: <ul style="list-style-type: none"> • Any breakpoints encountered are detected by the debugger • The process is halted when a breakpoint is encountered, and • The information is available in the Debug window 	
Detach From Process	To detach from a process, click on the  (Debug Stop) button.	Debugger Facilities ^[1874]

10.8.4.3.6 Process First Chance Exceptions

Access **Debug & Record Toolbar:**  | **Process First Chance Exceptions**

Topics

Topic	Detail	See also
Debug Process	When an application is being debugged and the debugger is notified of an exception, the application is paused and the debugger responds in the way it is configured to do; it either: <ul style="list-style-type: none"> • Resumes the application and leaves the exception to the application to manage, or • Keeps the application suspended and passes the exception to the appropriate routines for automatic resolution or manual intervention 	
Second Chance Exceptions	The Enterprise Architect debugger defaults to the first behavior, above. If the application can handle the exception, it continues to process; if it cannot handle the exception, the debugger is notified again and this time it must suspend the application and resolve the exception condition. In this behavior, because the debugger has encountered the exception twice,	

Topic	Detail	See also
	<p>it is known as a <i>second-chance exception</i>; in this case, if the exception does not halt execution, it is ignored and you avoid spending time on conditions that do not impact the overall outcome of processing.</p> <p>You might work this way on large or complex systems that invariably involve exception conditions somewhere in the processing paths.</p>	
First Chance Exceptions	<p>However, if you want to examine every exception that occurs as soon as it occurs, you can set the debugger to adopt the second behavior.</p> <p>Because the debugger responds to the exception on first contact, it is known as a <i>first-chance exception</i>.</p> <p>You might work this way with individual functions or routines that must work cleanly or not at all.</p>	
Selection	<p>Select the Process First Chance Exceptions option to debug exceptions on first contact.</p> <p>Deselect the option to process exceptions only if the application fails when they occur.</p>	

10.8.4.4 Run a Debug Session

10.9 Other Settings



You can set the default code options such as the editors for each of the programming languages available for Enterprise Architect and special options for how source code is generated or reverse engineered.

Learn more

Topic	Details	Link
General Options	Describes the general options that apply to all languages when generating code from Enterprise Architect.	Source Code Engineering Options ^[1897]
Local Paths	Describes how to define local paths for each Enterprise Architect user, using the Local Paths settings.	Local Paths ^[1900]
Local Path Dialog	Enables you to set up local paths for a single user on a particular machine.	Local Path Dialog ^[1901]
Language Macros	Describes how to define macros, which is helpful for reverse engineering.	Language Macros ^[1902]
Language Options	Describes the options for generating and reverse-engineering code in each of the supported languages.	Language Options ^[1907]
Set Collection Classes	Describes how to define Collection Classes.	Set Collection Classes ^[1904]

10.9.1 Source Code Engineering Options

The following topics describe general options that apply to all languages when generating code from Enterprise Architect. These options are all available under the Source Code Engineering section of the Options dialog.

Access [Tools | Options | Source Code Engineering](#)

Topic	Detail	Link
Source Code Options	You can set certain defined options for a particular language while generating the code.	Source Code Options ^[1892]

Topic	Detail	Link
Options - Code Editors	Describes how to access and configure the source code editor.	Options - Code Editors ^[1895]
Options - Object Lifetimes	Describes configuring various options concerning Object Lifetimes.	Options - Object Lifetimes ^[1895]
Options - Attributes/Operations	Describes configuring Attributes/Operations.	Options - Attribute/Operations ^[1899]
Code Page for Source Editing	Provides an option to define the Unicode character set for code generation.	Code Page for Source Editing ^[1900]

10.9.1.1 Source Code Options

When you generate code for a particular language, you can set certain options. These include:

- Create a default constructor
- Create a destructor
- Generate copy constructor
- Select default language
- Generate methods for implemented interfaces
- Set the Unicode options for code generation


Access [Tools | Options | Source Code Engineering](#)

Use to

- Configure code generation options

Reference

Field	Usage	See also
Always synchronize with existing file (recommended)	Indicates that imported code synchronizes with an existing file.	
Replace (overwrite) existing source file	Indicates that imported code overwrites an existing source file.	
Component Types	Allows for the importation of component types, this opens the Import component types dialog.	Import Component

		Types 
Default Language for Code Generation:	Indicates the default coding language used in code generation.	
Wrap long comment lines at:	Specifies the line length to wrap comments within.	
Auto Layout Diagram on Import:	Specifies if and when a diagram is automatically generated on code import.	
Output file use both CR & LF	Indicates if carriage return and line feed is a used; ensure that this option is set according to what operating system is currently in use as code might not render correctly.	
Prompt when synchronizing (reversing)	Ensures that a prompt displays when synchronization occurs.	
Remove hard breaks from comments on import	Indicates that hard breaks are removed from commented sections on importation.	
Auto generate role names when creating code	Indicates that role names are generated when creating code.	
Do not generate role names when creating code	Indicates that role names are not generated when creating code.	
Do not generate members where association direction is "Unspecified"	If association direction is "unspecified", members are not generated	
Create dependencies for operation returns and parameter types	Creates dependencies for operation returns and parameter types.	
Comments: Generate	Indicates that comments are generated.	
Comments: Reverse	Indicates that reverse comments are generated.	
Remove prefixes when generating Get/Set properties	Enables specification of prefixes used in your variable naming conventions, if those prefixes should be removed in the variables' corresponding get/set functions.	
Treat as suffixes	Treats the prefixes defined within Remove prefixes when	

	generating Get/Set properties as suffixes.	
Capitalized Attribute Name for properties	Capitalizes Attribute names for properties.	
Use "Is" for Boolean property Get()	Indicates that the <i>/s</i> keyword is used for the Boolean property Get() .	
Code page for source editing	Specifies the character embedding format.	

Notes

- It is worthwhile to configure these settings, as they serve as the defaults for all Classes in the model; you can override these on a per-Class basis using the custom settings (from the Code Generation dialog)

10.9.1.1.1 *Import Component Types*

The Import Component Types dialog enables you to configure what elements you would like to be created for files of any extension found while importing a source code directory.

Access [Tools | Options | Source Code Engineering : Component Types](#)

Use to

- Configure what elements are to be created for files of any extension found while importing a source code directory

Reference

Field	Usage	See also
Extension	Indicates the extension name for a component type.	
Type	Indicates the type for a component.	
Stereotype	Indicates the stereotype name given to a component.	
Component List	The defined list of component types.	
Save	Saves a component definition and adds it to the component list.	
New	Allows a new component type to be defined.	

Delete	Deletes a highlighted component from the component list.	

Notes

- You can transport these import component types between models, using the Export Reference Data and Import Reference Data options

Learn more

- [Export Reference Data](#)^[308]
- [Import Reference Data](#)^[308]

10.9.1.2 Options - Code Editors

You access the source code editor options via the Code Editors page of the Options dialog. They enable you to configure options for Enterprise Architect's internal editor, as well as the default editor for DDL scripts. You can configure external editors for code languages on each language options page.

Access [Tools | Options | Source Code Engineering > Code Editors](#)

Use to

- Configure options for Enterprise Architect's internal editor
- Configure the default editor for DDL Scripts
- Configure DDL Name Templates


Reference

Field	Usage	See also
DDL Editor	Defaults to blank, indicating that the Enterprise Architect code editor is the DDL editor in use. You can select a different default editor if necessary: click on the (...) button to browse for and select the the required DDL editor. The editor name then displays in the DDL Editor field.	
Default Database	Indicates the default database used.	
MySQL Storage Engine	Indicates the storage engine used.	
DDL Name Templates	Specifies the Foreign Key, Primary Key and Unique Constraint Name templates; the (...) button enables you	Define Primary Key Name Template ^[1954]

Field	Usage	See also
	to specify template names.	Define Foreign Key Name Template ^[1960]
Use inbuilt editor if no external editor set	Specifies the editor for code in a language if no external editor is defined for that language.	
Show Line Numbers	Shows line numbers in the editor.	
Show Structure Tree	Shows a tree with the results of parsing the open file (requires that the file is parsed successfully).	
Don't parse files larger than	Specifies an upper limit on file size for parsing. Use this to prevent performance decrease due to parsing very large files.	
Syntax Highlighting Options	Specifies both global and language-specific editor language properties.	Editor language properties ^[1896]
Automatically Reverse Engineer on File Save	When this is checked, pressing (Ctrl) + (S) to save in the source code editor automatically reverse engineers the code in the same way as the Save Source and Re-Synchronize Class button does.	

10.9.1.2.1 Editor Language Properties

Enterprise Architect enables you to specify syntax highlighting properties for all the programming languages that Enterprise Architect supports at installation.

Access **Tools | Options | Source Code Engineering > Code Editors : Syntax Highlighting Options (...)**
Project Browser element context menu | View Source Code:  **| Syntax Highlighting Options**

Use to

- Configure Syntax highlighting for various coding languages

Reference

On the Editor Language properties dialog:

Topic	Detail	See also
General Features	<p>The panel on the left of the dialog lists the languages for which you can set properties.</p> <p>The <i>(Global)</i> item at the top of the list enables you to set properties that apply to all programming languages; however, you can reset a global property to a different value for a specific language, on the page for that language.</p> <p>Resetting a global property for one language does not affect that property's value for the other languages.</p> <p>Click on the required language in the list, to display the properties for that language:</p> <ul style="list-style-type: none"> • Properties shown in bold indicate that this is the highest level at which this property can be defined (for most language options other than <i>Global</i>, this is effectively the only point at which the property is defined) • Properties shown in normal font are generally the global properties that you can reset just for the current language <p>Scroll through the property categories and individual properties for the language; you can collapse and expand categories as necessary, using the expansion box next to the category name (☐).</p> <p>When you click on a property name, an explanation of that property displays in the panel at the bottom right of the dialog.</p>	
Define Properties	<p>To define a property, click on the value field following the property name; depending on the type of property, either the field is enabled for direct editing or a drop-down arrow or Browse button displays (as described for the Tagged Values window) to enable you to select the values to define the property.</p> <p>Select or type in the required values.</p> <p>The Toolbar icons enable you to:</p> <ul style="list-style-type: none"> • Save your changes to the properties • Reset ALL properties fields to the default settings shipped with Enterprise Architect • Reset the current style field to the default setting (not enabled for non-style fields) 	Tagged Values [973]
Assign Keys to Macros	<p>The Macros category enables you to assign (Ctrl+Alt+n) keystroke combinations to coding macros that you have created yourself in the Source Code Viewer.</p> <p>When you click on the Browse button in a selected Macro field, the Open Macro dialog displays; this dialog lists the existing macros and, if a key combination has been assigned to a macro, what that key combination is.</p>	Source Code Viewer [1802]

Topic	Detail	See also
	Click on the name of the macro and on the Open button to assign the selected keys to the macro.	

Notes

- You cannot currently set properties for any additional languages you include through an MDG Technology
- You can resize this dialog, if required

Learn more

- [Function Details](#)^[1810] (Syntax Highlighting)
- [MDG Technology](#)^[1331]

10.9.1.3 Options - Object Lifetimes

This set of options enables you to configure various options concerning Object Lifetimes.

Access [Tools | Options | Source Code Engineering > Object Lifetimes](#)

Use to

- Define constructor details when generating code
- Specify whether to create a copy constructor
- Destructor details

Reference

Field	Usage	See also
Constructor	This set of options specify if a constructor is generated, if a constructor is in-line and the visibility of the default constructor.	
Copy Constructor	This set of options specify if a copy constructor is generated, if a copy constructor is in-line and the visibility of the default copy constructor.	
Destructor	This set of options specify if a destructor is generated, if a destructor is in-line, if a destructor is defined as <i>virtual</i> and the visibility of the default constructor.	

Notes

- Some options are related to C++ only

10.9.1.4 Options - Attribute/Operations


The Options - Attributes/Operations section enables you to configure Attribute and Operations specifications.

Access [Tools | Options | Source Code Engineering > Attribute/Operations](#)

Use to

- Configure the default name generated from imported attributes
- Generate methods for implemented interfaces
- Delete model attributes not included in the code during reverse synchronization
- Delete model methods not included in the code during reverse synchronization
- Delete code from features contained in the model during forward synchronization
- Delete model associations and aggregations that correspond to attributes not included in the code during reverse synchronization
- Define whether or not the bodies of methods are included and saved in the Enterprise Architect model when reverse engineering
- Create attributes in quick succession, clearing the dialog when you click on **Save** so that you can enter another attribute name

Reference

Field	Usage	See also
Attribute Specifications: Default name for associated	Specifies the default name generated from imported attributes.	
Attribute Specifications: On reverse synch, delete model attributes not in code	Indicates that on reverse synchronization, attributes in the model that are not included within code are automatically removed from the model.	Update Package Contents 
Attribute Specifications: On reverse synch, delete model associations not in code	Indicates that on reverse synchronization, associations in the model that are not included within code are automatically removed from the model.	
Operation Specifications: Generate methods for implemented interface	Indicates that methods are generated for an implemented interface	
Operation Specifications: On reverse synch, delete model	Indicates that on reverse synchronization, methods in the model that are not included within code are	Update Package

Field	Usage	See also
methods not in code	automatically removed from the model.	Contents ^[1774]
Operation Specifications: Include method bodies in model when reverse engineering	Indicates that on reverse engineering code, method bodies in the code are included within your model.	
Options: After Save, re-select edited item	Indicates that after saving, the dialog clears allowing for another attribute name to be specified.	
On forward synch, prompt to delete code features not in mode	Indicates that, during forward synchronization, the Synchronize Element <package name>.<element name> dialog displays, to enable to to either ignore, reassign or delete features in the code that are not in the model.	Synchronize Model and Code ^[1775]

10.9.1.5 Code Page for Source Editing

Enterprise Architect enables you to define the Unicode character set for code generation.

Access [Tools](#) | [Options](#) | [Source Code Engineering](#)

How to

To set the Unicode character set

Step	Action	See also
1	Select the Source Code Engineering option. The Source Code Engineering page of the Options dialog displays.	
2	In the Code page for source editing field, click on the drop-down arrow and select the appropriate Unicode character set.	
3	Click on the Close button.	

10.9.2 Local Paths

Access [Settings](#) | [Local Directories and Paths](#)

Topics

Topic	Detail	See also
General Features	<p>Sometimes a team of developers could be working on the same Enterprise Architect model; each developer might store their version of the source code in their local file system, but not always at the same location as their fellow developers.</p> <p>To handle this scenario, Enterprise Architect enables you to define local paths for each Enterprise Architect user, using the Local Paths dialog (select the Local Directories and Paths menu option).</p> <p>You can use local paths in generating code and reverse engineering, and in version control, developing XML schemas and generating RTF and HTML reports.</p> <p>Local paths take a bit of setting up, but if you want to work collaboratively on source and model concurrently, the effort is well worth while.</p> <p>For example:</p> <ul style="list-style-type: none"> Developer A stores her .java files in a <code>C:\Java\Source</code> directory, while developer B stores his in <code>D:\Source</code> Both developers want to generate and reverse engineer into the same Enterprise Architect model located on a shared (or replicated) network drive <p>Developer A might define a local path of:</p> <pre>JAVA_SOURCE = " C: \ J a v a \ S o u r c e "</pre> <p>All Classes generated and stored in the Enterprise Architect project are stored as:</p> <pre>%J A V A _ S O U R C E % \ < x x x . j a v a ></pre> <p>Developer B defines a local path as:</p> <pre>J A V A _ S O U R C E = " D : \ S o u r c e "</pre> <p>Now, Enterprise Architect stores all java files in these directories as:</p> <pre>%J A V A _ S O U R C E % \ < f i l e n a m e ></pre> <p>On each developer's machine, the filename is expanded to the correct local version.</p>	Local Paths Dialog

10.9.3 Local Paths Dialog

The Local Paths dialog enables you to set up local paths for a single user on a particular machine. For a description of what Local Paths are used for, see the *Local Paths* topic.

Access [Settings | Local Directories and Paths](#)

Topics

Topic	Detail	See also
General Usage	<p>The Local Paths dialog enables you to define:</p> <ul style="list-style-type: none"> • Path - the local directory in the file system (for example, d:\j a v a \ s o u r c e) • ID - the shared ID that is substituted for the Local Path (for example, J A V A _ S R C) • Type - the language type (for example, Java) <p>And also:</p> <ul style="list-style-type: none"> • Apply Path - select a path and click on this button to update any existing paths in the model (with full path names) to the shared relative path name (for example: d: \ j a v a \ s o u r c e \ m a i n . j a v a might become % J A V A _ S R C % \ m a i n . j a v a) • Expand Path - the opposite of Apply Path; this enables you to remove a relative path and substitute the full path name <p>Using the two options you can update and change existing paths.</p>	

Notes

- You can also set up a hyperlink on a diagram to access the Local Paths dialog, to switch, update or expand your current Local Path

Learn more

- [Local Paths](#)^[1900]
- [Hyperlink](#)^[1668]

10.9.4 Language Macros

When reverse engineering a language such as C++, you might find preprocessor directives scattered throughout the code. This can make code management easier, but can hamper parsing of the underlying C++ language.

To help remedy this, you can include any number of *macro* definitions, which are ignored during the parsing phase of the reverse engineering. It is still preferable, if you have the facility, to preprocess the code using the appropriate compiler first; this way, complex macro definitions and defines are expanded out and can be readily parsed. If you don't have this facility, then this option provides a convenient substitute.

Access [Settings | Preprocessor Macros](#)

How to

To define a macro

Step	Action	See also
1	Select the Preprocessor Macros menu option. The Language Macros dialog displays.	
2	Click on the Add New button.	
3	Enter details for your macro.	
4	Click on the OK button.	

Topics

Topic	Detail	See also
Macros Embedded Within Declarations	<p>Macros are sometimes used within the declaration of Classes and operations, as in the following examples:</p> <pre>class __declspec Foo { int __declspec Bar (int p); };</pre> <p>If <i>declspec</i> is defined as a C++ macro, as outlined above, the imported Class and operation contain a Tagged Value called <i>DeclMacro1</i> with value <i>__declspec</i> (subsequent macros would be defined as <i>DeclMacro2</i>, <i>DeclMacro3</i> and so on).</p> <p>During forward engineering, these Tagged Values are used to regenerate the macros in code.</p>	
Define Complex Macros	<p>It is sometimes useful to define rules for complex macros that can span multiple lines; Enterprise Architect ignores the entire code section defined by the rule.</p> <p>Such macros can be defined in Enterprise Architect as in the following two examples; both types can be combined in one definition.</p> <p>Block Macros</p> <pre>BEGIN_INTERFACE_PART ^ END_INTERFACE_PART</pre> <p>The ^ symbol represents the body of the macro - this enables skipping from one macro to another; the spaces surrounding the ^ symbol are required.</p> <p>Function Macros</p>	

Topic	Detail	See also
	<p>RTTI _EMULATI ON()</p> <p>Enterprise Architect skips over the token including everything inside the parentheses.</p>	

Notes

- You can transport these language macro (or preprocessor macro) definitions between models, using the **Export Reference Data** and **Import Reference Data** options; the macros are exported as a Macro List

Learn more

- [Export Reference Data](#)^[306]
- [Import Reference Data](#)^[308]

10.9.5 Set Collection Classes

Enterprise Architect enables you to define *Collection Classes* for generating code from Association connectors where the target role has a multiplicity setting greater than 1.

Topic	Detail	See also
Defining Collection Classes	<p>On the Source Code Engineering section of the Options dialog (select the Tools Options Source Code Engineering option), on each language page click on the Collection Classes button.</p> <p>The Collection Classes for Association Roles dialog displays.</p> <p>On this dialog, you can define:</p> <ul style="list-style-type: none"> • The default Collection Class for 1..* roles • The ordered Collection Class to use for 1..* roles • The qualified Collection Class to use for 1..* roles • 	<p>Source Code Options^[1892]</p>
Defining Collection Classes for a specific Class	<p>Class-specific Collection classes can be defined by clicking the Collection Classes button in the Class Properties dialog of the element.</p>	<p>Element Details^[846]</p>
Code Generation Precedence	<p>When Enterprise Architect generates code for a connector that has a multiplicity role >1:</p> <ol style="list-style-type: none"> 1. If the Qualifier is set use the qualified collection: <ul style="list-style-type: none"> • for the Class if set 	




Topic	Detail	See also
	<ul style="list-style-type: none"> • else use the code language qualified collection <p>2. If the Order option is set use the ordered collection:</p> <ul style="list-style-type: none"> • for the Class if set • else use the code language ordered collection <p>3. Else use the default collection:</p> <ul style="list-style-type: none"> • for the Class if set • else use the code language default collection 	
Using Markers	<p>You can include the marker #TYPE# in the collection name; Enterprise Architect replaces this with the name of the Class being collected at source generation time (for example, <i>Vector<#TYPE#></i> would become <i>Vector<foo></i>).</p> <p>Conversely, when reverse engineering, an Association connector is also created if a matching entry (e.g. <i>foo</i> if <i>foo</i> is found in the model) is defined as a Collection Class.</p>	
Additional Collection Classes	Additional Collection Classes can be defined in the language options page (C#, C++, Java).	Example Use of Collection Classes C# Options C++ Options Java Options
Member Type	<p>On both the Source Role and Target Role tabs of the Association Property dialog (accessible from the right-click context menu of any Association) there is a Member Type field.</p> <p>If you set this, the value you enter overrides all the above options.</p>	Source Role Target Role

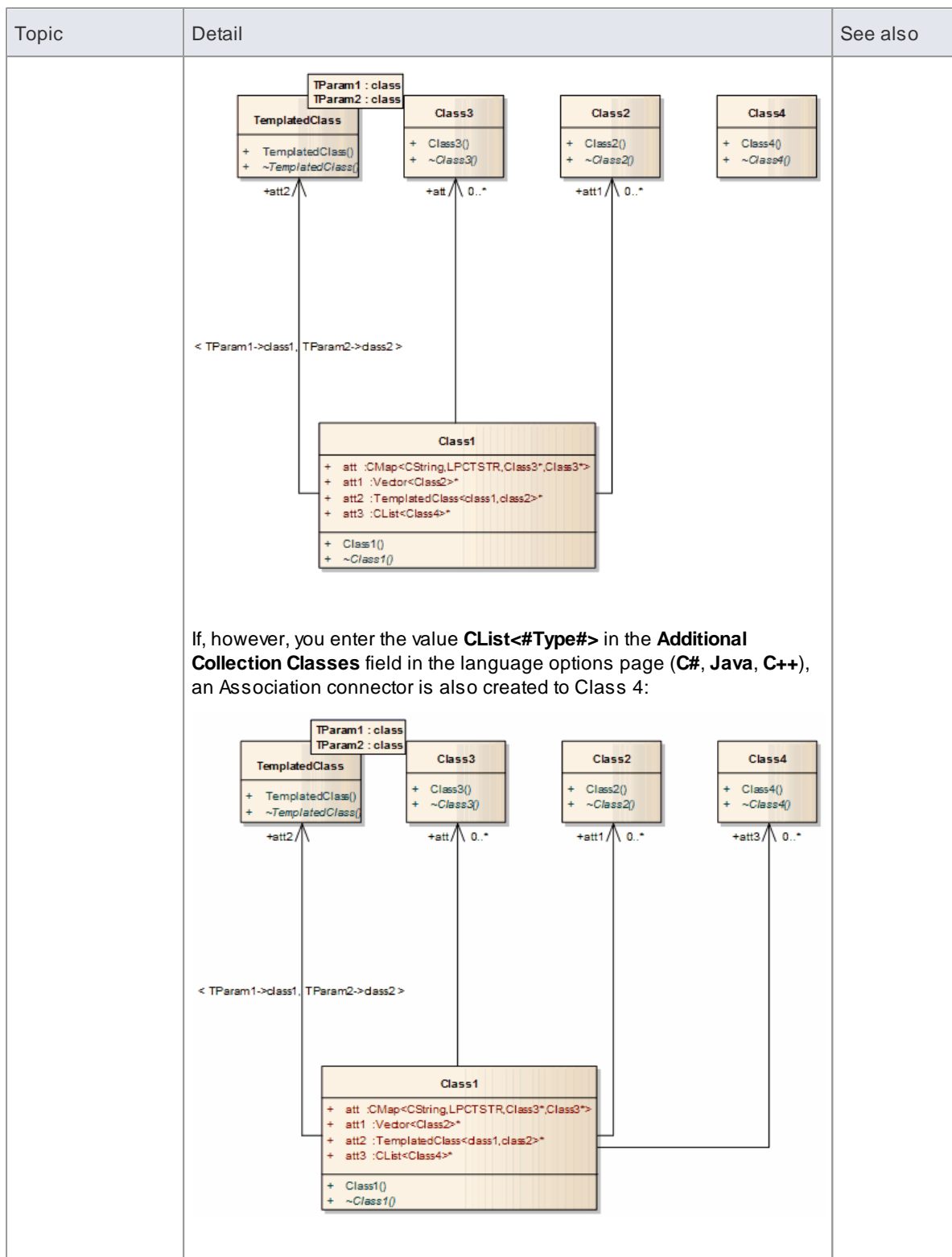
Learn more

- [Generate Source Code](#)
- [Importing Source Code](#)
- [Source Code Options](#)

10.9.5.1 Example Use of Collection Classes

Topics

Topic	Detail	See also
Example Code	<p>Consider the following source code:</p> <pre> class Class1 { public: Class1(); virtual ~Class1(); CMap<CString, LPCTSTR, Class3*, Class3*> att; Vector<Class2> *att1; TemplatedClass<class1, class2> *att2; List<Class4> *att3; }; class Class2 { public: Class2(); virtual ~Class2(); }; class Class3 { public: Class3(); virtual ~Class3(); }; class Class4 { public: Class4(); virtual ~Class4(); }; template<class TParam1, class TParam2> class TemplatedClass { public: TemplatedClass() { } virtual ~TemplatedClass() { } }; </pre>	
Generated Diagram	<p>If this code above is imported into Enterprise Architect with default import options, the following diagram is generated:</p>	<p>C# Options </p> <p>Java Options </p> <p>C++ Options </p>



10.9.6 Language Options

You can set up various options for how Enterprise Architect handles a particular language when generating and reverse-engineering code. These options are accessible on the Options dialog (select the **Tools | Options** menu option).

Under the **Source Code Engineering** option, select the required language. The following topics outline the

options available for each language.

Topic	Link
Action Script	ActionScript ^[1909]
Ada 2005 (in the Systems Engineering and Ultimate editions of Enterprise Architect)	Ada 2005 ^[1909]
ANSI C	ANSI C ^[1911]
C#	C# ^[1912]
C++	C++ ^[1912]
Delphi	Delphi ^[1913]
Delphi Properties	Delphi Properties ^[1914]
Java	Java ^[1915]
PHP	PHP ^[1916]
Python	Python ^[1917]
SystemC	SystemC ^[1917]
Verilog (Systems Engineering and Ultimate editions)	Verilog ^[1919]
VHDL (Systems Engineering and Ultimate editions)	VHDL ^[1919]
Visual Basic	Visual Basic ^[1920]
Visual Basic .NET	Visual Basic .NET ^[1918]
MDG Technology Languages	MDG Technology Languages ^[1921]
Reset Options	Reset Options ^[1922]

Topic	Link

10.9.6.1 ActionScript Options

Configure options for ActionScript code generation using the ActionScript Specifications page of the Options dialog.

Access [Tools | Options | Source Code Engineering > ActionScript](#)

Use to

- Specify default ActionScript version to generate (AS2.0 or AS3.0)
- Specify default file extensions (header and source)
- Specify default source directory
- Specify the editor for ActionScript code

Reference

Field	Usage	See also
Disable Language	Indicates that ActionScript language support is to be disabled.	
Options for the current model	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions.	Collection Classes 1904

10.9.6.2 Ada 2005 Options

Configure options for Ada 2005 code generation using the Ada page of the Options dialog.

Access [Tools | Options | Source Code Engineering > Ada](#)

Use to

- Inform the reverse engineering process whether the name of the Tagged Record is the same as the package name

- Advise the engine of the alternate Tagged Record name to locate
- Specify whether the engine should create a reference type for the Tagged Record (if one is not defined)
- Supply the name of the reference type to be created (default is Ref)
- To specify the reference parameter of a Reference / Access type
- Tell the engine to ignore the name of the reference parameter
- Indicate the name of the reference parameter to locate

Reference

Field	Usage	See also
Disable Language	Indicates that Ada language support is to be disabled.	
Options for the current model	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions.	Collection Classes 1904

Notes

- Ada 2005 support is available in the System Engineering and Ultimate editions of Enterprise Architect

10.9.6.3 ArcGIS Options

Configure options for ArcGIS code generation using the ArcGIS page of the Options dialog.

Access **Tools | Options | Source Code Engineering > ArcGIS**

Use to

- Specify default file extensions
- Specify default source directory
- Specify the editor for ArcGIS code

Reference

Field	Usage	See also
-------	-------	----------

Disable Language	Indicates that ArcGIS language support is to be disabled.	
Options for the current model	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions.	Collection Classes [1904]

10.9.6.4 C Options

Configure options for C code generation using the C Specifications page of the Options dialog.

Access [Tools | Options | Source Code Engineering > C](#)

Use to

- Support Object Oriented coding
- Indicate default file extensions (header and source)
- Indicate default source directory
- Indicate an editor for C code
- Specify a path that Enterprise Architect uses to search for the implementation file; the first path in the list is the default path when generating

Reference

Field	Usage	See also
Disable Language	Indicates that C language support is to be disabled.	
Options for the current model	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions.	Collection Classes [1904]

--	--	--

10.9.6.5 C# Options

Configure options for C# code generation using the C# Specifications page of the Options dialog.

Access [Tools | Options | Source Code Engineering > C#](#)

Use to

- Indicate the default File extension
- Indicate additional Collection Classes - to define custom collection classes, which can be simple substitutions (such as *CArray<#TYPE#>*) or a mix of other strings and substitutions (such as *Cmap<CString,LPCTSTR,#TYPE#*,#TYPE#*>*); the following collection classes are defined by default:
 - *List<#TYPE#>*; *Stack<#TYPE#>*; *Queue<#TYPE#>*;
- Specify a 'Get' prefix
- Specify a 'Set' prefix
- Specify a directory for opening and saving C# source code

Reference

Field	Usage	See also
Disable Language	Indicates that C# language support is to be disabled.	
Options for the current model	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions.	Collection Classes 1904

10.9.6.6 C++ Options


Configure options for C++ code generation using the C++ Specifications page of the Options dialog.

Access [Tools | Options | Source Code Engineering > C++](#)

Use to

- Indicate the version of C++ to generate; this controls the set of templates used and how properties are created
- Specify the default reference type used when a type is specified by reference
- Specify the default file extensions
- Specify default Get/Set prefixes
- Define additional Collection Classes - to define custom collection classes, which can be simple substitutions (such as *CArray<#TYPE#>*) or a mix of other strings and substitutions (such as *Cmap<CString,LPCTSTR,#TYPE#*,#TYPE#*>*); the following collection classes are defined by default:
 - *CArray<#TYPE#>;CMap<CString,LPCTSTR,#TYPE#*,#TYPE#*>;*
- Specify a default source directory
- Specify the path that Enterprise Architect uses to search for the implementation file; the first path in the list is the default path when generating new implementation files and parsing existing files - if you add further directories, Enterprise Architect also searches these when parsing existing files

Reference

Field	Usage	See also
Disable Language	Indicates that C++ language support is to be disabled.	
Options for the current model	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions.	Collection Classes 

10.9.6.7 Delphi Options

Configure options for Delphi code generation using the Delphi Specifications page of the Options dialog.

Access [Tools | Options | Source Code Engineering > Delphi](#)

Use to

- Indicate a default file extension
- Indicate a default source directory
- Set a default directory for opening and saving Delphi source code

Reference

Field	Usage	See also
Disable Language	Indicates that Delphi language support is to be disabled.	
Options for the current model	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions.	Collection Classes ^[1904]

Learn more

- [Delphi Properties](#) ^[1914]

10.9.6.7.1 Delphi Properties

Enterprise Architect has comprehensive support for Delphi properties. These are implemented as Tagged Values, with a specialized property editor to help create and modify Class properties. By using the **Feature Visibility** element context menu option, you can display the 'tags' compartment that contains the properties. Imported Delphi Classes with properties have this feature automatically made visible for your convenience.

Use to

- Configure additional properties for Delphi based elements

Reference

Topic	Description	See also
To manually activate the property editor	<ul style="list-style-type: none"> • Ensure the Class you have selected has the code generation language set to Delphi • Right-click on the Class and select the Delphi Properties context menu option to open the editor <p>The Delphi Properties editor enables you to build properties in a simple and straightforward manner; from here you can:</p> <ul style="list-style-type: none"> • Change the name and scope (only Public and Published are currently supported) 	

	<ul style="list-style-type: none"> • Change the property type (the drop-down list includes all defined Classes in the project) • Set the Read and Write information (the drop-down lists have all the attributes and operations from the current Class; you can also enter free text) • Set Stored to True or False • Set the Implements information • Set the default value, if one exists 	
--	--	--

Notes

- When you use the Create Property dialog from the Attribute screen, Enterprise Architect generates a pair of Get and Set functions, together with the required property definition as Tagged Values; you can manually edit these Tagged Values if required
- Public properties are displayed with a '+' symbol prefix and published with a '^'
- When creating a property in the Create Property Implementation dialog (accessed through the Attributes dialog), you can set the scope to Published if the property type is Delphi
- Only **Public** and **Published** are supported
- If you change the name of a property and forward engineer, a new property is added, but the old one must be manually deleted from the source file

10.9.6.8 Java Options

Configure options for Java code generation using the Java Specifications page of the Options dialog.

Access [Tools | Options | Source Code Engineering > Java](#)

Use to

- Set a default directory for opening and saving Java source code
- Indicate a default file extension
- Specify a default 'Get' prefix
- Specify a default 'Set' prefix
- Define additional Collection Classes - to define custom collection classes, which can be simple substitutions (such as *CArray<#TYPE#>*) or a mix of other strings and substitutions (such as *Cmap<CString,LPCTSTR,#TYPE#*,#TYPE#*>*); the following collection classes are defined by default:
 - *HashSet<#TYPE#>;Map<String,#TYPE#>;*

Reference

Field	Usage	See also
-------	-------	----------

Disable Language	Indicates that Java language support is to be disabled.	
Options for the current model	Specifies the options used in the current model. These options affect all users of the current model; however they do not apply to other models.	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions.	Collection Classes <small>1904</small>

10.9.6.9 PHP Options

Configure options for PHP code generation using the PHP Specifications page of the Options dialog.

Access [Tools | Options | Source Code Engineering > PHP](#)

Use to

- Specify the extension to be used when creating files for PHP source
- Define a semi-colon separated list of extensions to look at when doing a directory code import for PHP
- Set a default directory for opening and saving PHP source code
- Specify the version of PHP to generate

Reference

Field	Usage	See also
Disable Language	Indicates that PHP language support is to be disabled.	
Options for the current model	Specifies the options used in the current model. These options affect all users of the current model; however they do not apply to other models.	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	

Learn more

- [Import a Directory Structure](#)
1795

10.9.6.10 Python Options

Configure options for Python code generation using the Python Specifications page of the Options dialog.

Access [Tools](#) | [Options](#) | [Source Code Engineering](#) > [Python](#)

Use to

- Indicate the default file extension(s)
- Specify the default source directory to be used
- Specify the editor used to write code

Reference

Field	Usage	See also
Disable Language	Indicates that Python language support is to be disabled.	
Options for the current model	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	

10.9.6.11 SystemC Options

Configure options for SystemC code generation using the SystemC page of the Options dialog.

Access [Tools](#) | [Options](#) | [Source Code Engineering](#) > [SystemC](#)

Use to

- Indicate default file extension(s)
- Specify a default source directory
- Specify an editor for changing code

Reference

Field	Usage	See also
Disable Language	Indicates that SystemC language support is to be disabled.	

Options for the current model	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions.	Collection Classes [1904]

10.9.6.12 VB.NET Options

Configure options for VB.NET code generation using the VB.NET Specifications page of the Options dialog.

Access [Tools](#) | [Options](#) | [Source Code Engineering](#) > [VB.NET](#)

Use to

- Indicate default file extension(s)
- Specify a default source directory
- Specify an editor for changing code

Reference

Field	Usage	See also
Disable Language	Indicates that VB.NET language support is to be disabled.	
Options for the current model	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions.	Collection Classes [1904]

10.9.6.13 Verilog Options

Configure options for Verilog code generation using the Verilog page of the Options dialog.

Access [Tools | Options | Source Code Engineering > Verilog](#)

Use to

- Indicate default file extension(s)
- Specify a default source directory
- Specify an editor for changing code

Reference

Field	Usage	See also
Disable Language	Indicates that Verilog language support is to be disabled.	
Options for the current model	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions.	Collection Classes ^[1904]

10.9.6.14 VHDL Options


Configure options for VHDL code generation using the VHDL page of the Options dialog.

Access [Tools | Options | Source Code Engineering > VHDL](#)

Use to

- Indicate default file extension(s)
- Specify a default source directory
- Specify an editor for changing code

Reference

Field	Usage	See also
Disable Language	Indicates that VHDL language support is to be disabled.	
Options for the current model	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions.	Collection Classes 

10.9.6.15 Visual Basic Options

Configure options for Visual Basic code generation using the VB Specifications page of the Options dialog.

Access [Tools | Options | Source Code Engineering > Visual Basic](#)

Use to

- Indicate the default file extension when reading/writing
- Specify the default Visual Basic version
- Indicate the MTS transaction mode for MTS objects
- Specify if a Class uses Multi use (**true** or **false**)
- Specify if a Class uses the **Persistable** property
- Indicate data binding behaviours
- Set the global namespace
- Set the *Exposed* attribute
- Indicate if the *Creatable* attribute is **true** or **false**

Reference

Field	Usage	See also
Disable Language	Indicates that Visual Basic language support is to be disabled.	
Options for the current model	Specifies the options used in the current model. These options affect all users of the current model; however,	

	they do not apply to other models.	
Options for the current user	Specifies the options used for the current use; these options apply to all models that are accessed by the user.	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions.	Collection Classes

10.9.6.16 MDG Technology Language Options

If you have loaded an MDG Technology that specifies a code module into your *Sparx Systems > EA > MDG Technologies* folder, the language is included in the Source Code Engineering list on the Options dialog. The language is only listed on the Options dialog if an MDG Technology file actually uses it in your model.

Access [Tools | Options | Source Code Engineering > MDG](#)

Reference

Field	Usage	See also
Default Extension	Default extension for generated source files; shown if the option is in the technology. This is saved per project.	
Import File Extensions	Default folder to import source files from; shown if the technology supports namespaces. This is saved once for all projects.	
Generate Namespaces	Indicates if namespaces are generated or not.	
Default Source Directory	The default directory to save generated source files. This is always shown.	
Editor	Indicates the editor that is used to edit source files.	
Att Type	Indicates the default attribute type.	

Notes

- These options are set in the technology inside the `<CodeOptions>` tag of a code module, as follows:

```
<CodeOption name="DefaultExtension" >.rb</CodeOption>
```

Learn more

- [MDG Technology](#)^[1323]
- [Adding Code Modules in MDG Technologies](#)^[1337]

10.9.6.17 Reset Options

Enterprise Architect stores some of the options for a Class when it is first created. Some are global; for example, *\$LinkClass* is stored when you first create the Class, so it won't automatically pick up the global change in the Options dialog in existing Classes. You must modify the options for the existing Class.

How to

To modify options for a single Class

Step	Action	See also
1	Right-click on the Class to change, and select the Generate Code context menu option. The Generate Code dialog displays.	
2	Click on the Advanced button. The Object Options dialog displays.	
3	Click on the Attributes/Operations button.	
4	Change the options, and click on the Close button to apply the changes.	

To modify options for all Classes within a package

Step	Action	See also
1	Right-click on the package in the Project Browser. The context menu displays.	
2	Select the Code Engineering Reset Options for this Package menu option. The Manage Code Generation dialog displays.	
3	Reset the required defaults for each existing Class.	

Step	Action	See also
4	Click on the OK button to apply the changes.	

10.10 Code Template Framework

The Code Template Framework (CTF) is used during forward engineering of UML models. The CTF enables you to:

- Generate source code from UML models
- Customize the way in which Enterprise Architect generates source code
- Forward engineer languages not specifically supported by Enterprise Architect

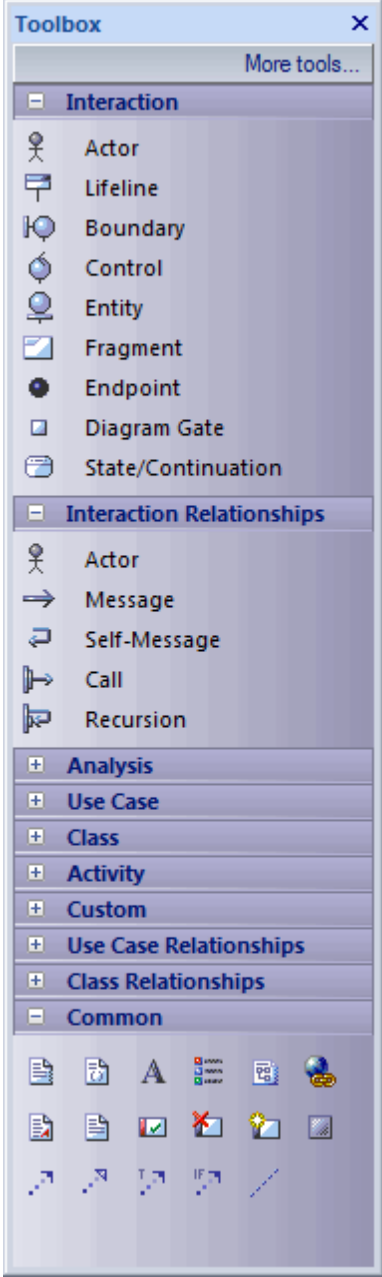
Topics

Topic	Detail	See also
Default Templates	Default Code Templates are built into Enterprise Architect for forward engineering supported languages.	Code and Transform Templates ^[1395] Base Templates ^[1397]
Code Template Editor	A Code Template Editor is provided for creating and maintaining user-defined Code Templates.	Code Template Editor ^[1404]
Customizing Code Templates	Descriptions of the template syntax and the macros and functions you can use to control the effects of the templates.	Code Template Framework ^[1394]
Synchronize Code	A subset of the default Code Templates to synchronize code.	Synchronize Code ^[1401]

10.11 ICONIX

Topics

Topic	Detail	See also
Abstract	<p>The following text is derived from the ICONIX entry in the online Wikipedia:</p> <p><i>The ICONIX Process is a minimalist, streamlined approach to Use Case driven UML modeling that uses a core subset of UML diagrams and techniques to provide thorough coverage of object-oriented analysis and design. Its main activity is robustness analysis, a method for bridging the gap between analysis and design. Robustness analysis reduces the ambiguity in use case descriptions, by ensuring that they are written in the context of an accompanying domain model. This process makes the use cases much easier to design, test and estimate.</i></p> <p>The ICONIX Process was developed by Doug Rosenberg; for more information on ICONIX, refer to <i>ICONIX Software Engineering Inc.</i></p>	ICONIX (Online Resource)
ICONIX in Enterprise Architect	<p>Enterprise Architect enables you to develop models under ICONIX quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer.</p> <p>The ICONIX facilities are provided in the form of:</p> <ul style="list-style-type: none"> • A set of ICONIX pages in the Toolbox • ICONIX element and relationship entries in the Toolbox Shortcut Menu and Quick Linker <p>To further help you develop and manage a project under ICONIX, Enterprise Architect also provides a white paper on the ICONIX Roadmap.</p> <p>In addition, Enterprise Architect has an alternative visual layout specific to ICONIX.</p>	Toolbox Shortcut Menu ^[704] Quick Linker ^[791] ICONIX Roadmap (Online Resource)
ICONIX Toolbox Pages	<p>Within the Toolbox, Enterprise Architect provides ICONIX versions of the pages for UML Analysis, Use Case, Class, Interaction (Sequence), Activity and Custom diagrams (which often form the basis for <i>Robustness diagrams</i>).</p> <p>Compared to the standard Toolbox pages, these have slightly different element and relationship sets; you can access them by either:</p> <ul style="list-style-type: none"> • Selecting the More tools ICONIX <Diagram Type> menu option for a specific Toolbox page • Selecting the ICONIX option in the drop-down field of the Default Tools toolbar, which adds all six pages to the Toolbox; the first page and the Common page are 	Analysis Diagram ^[1509] Use Case Diagram ^[1030] Class Diagram ^[1014] Sequence Diagram ^[1074] Activity Diagram ^[1028] Custom Diagram ^[1505]

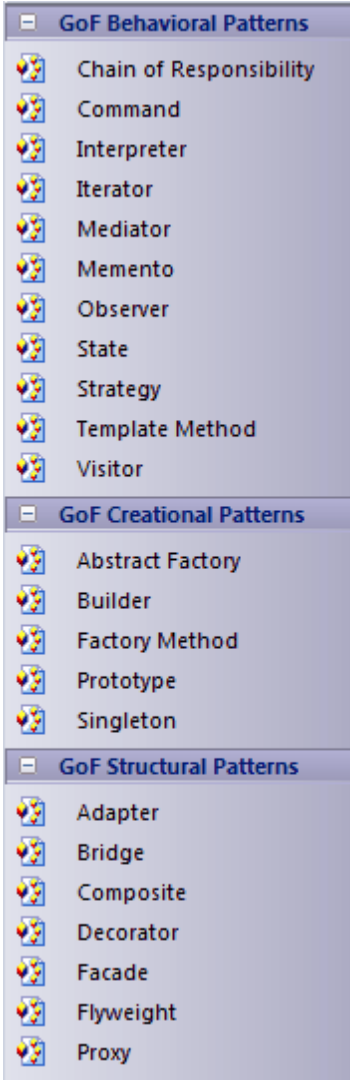
Topic	Detail	See also
	<p>expanded, and the others are closed up</p> 	
<p>ICONIX Layout</p>	<p>The ICONIX layout re-organizes the Enterprise Architect work area, opening the:</p> <ul style="list-style-type: none"> • Toolbox on the right hand side of the screen (follow the instructions above to display the ICONIX pages) • The Learning Center window auto-hidden in the top right of the screen 	<p>Workspace Layout [149]</p>

Topic	Detail	See also
	<ul style="list-style-type: none">• Project Browser window in the top left of the screen, and• Notes, Properties and Tagged Values windows nested on the bottom left of the screen <p>To apply this layout, select the View Workspace Layouts Basic Layout (Alternate) menu option.</p>	
Disable ICONIX	<p>If you prefer not to use ICONIX in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).</p> <p>This does not affect the ICONIX layout, which you can switch back to your own layout or the Enterprise Architect default layout using the View Workspace Layouts menu option.</p>	MDG Technologies <small>1285</small>

10.12 GoF Patterns

Topics

Topic	Detail	See also
Abstract	<p>The following text is derived from the <i>Design Patterns</i> entry in the online Wikipedia:</p> <p><i>Gang of Four (GoF) Patterns are 23 classic software design patterns providing recurring solutions to common problems in software design. They were developed by Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, often referred to as the Gang of Four. The patterns are defined in the book Design Patterns: Elements of Reusable Object-Oriented Software (Gamma et al., ISBN 0-201-63361-2).</i></p> <p>For further information on the concepts of GoF Patterns, refer to the <i>Wikipedia</i> item and its linked sources.</p>	Design Patterns (Online Resource)
GoF Patterns in Enterprise Architect	<p>Enterprise Architect enables you to develop diagrams from GoF patterns quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer.</p> <p>The GoF Patterns are provided in the form of:</p> <ul style="list-style-type: none"> GoF Behavioral Patterns, GoF Creational Patterns and GoF Structural Patterns pages in the Toolbox Gang of Four pattern entries in the Toolbox Shortcut Menu <p>GoF Toolbox Pages</p> <p>You can access the GoF Pattern pages of the Toolbox through the More tools GoF Patterns menu option; the following icons are available:</p>	Toolbox Shortcut Menu ^[704] Use A Pattern ^[1274]

Topic	Detail	See also
	 <p>When you drag one of the pattern elements onto a new diagram, the Add Pattern GoF <pattern group><pattern type> dialog displays; if necessary, modify the action and/or default for the component elements, then click on the OK button to create a diagram based on the pattern.</p> <p>The GoF patterns are drawn from the Resources window.</p> <p>If you delete a pattern in the Resources window the equivalent Toolbox item cannot work; therefore, if you cannot drop a pattern element from the Toolbox, check that it is still available in the Resources window.</p>	
Disable GoF Patterns	<p>If you prefer not to use the GoF Patterns technology in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).</p>	<p>MDG Technologies 1285</p>

Part

XI

11 Database Engineering



Enterprise Architect supports Data Modeling on different levels of abstraction to develop the standard Conceptual, Logical and Physical models. It supports:

- Interconnecting models across the three levels
- Using MDA transforms to generate the Physical model from the Logical model
- The ability to generate schema scripts from the Physical model for loading to a DBMS
- Features for remodeling legacy systems, including Reverse Engineering an existing DBMS structure

This section describes database engineering, explaining:

- How to create a Physical Data Model
- How to import database schema, and
- How to generate DDL for the model

Learn more

- [Data Models](#)^[1625] (Conceptual, Logical and Physical Models)
- [Import Database Schema](#)^[1973]
- [Generate DDL](#)^[1978]
- [Physical Data Model](#)^[1932]
- [Supported Databases](#)^[1970]

Learning Center topics

- **(Alt + F1) | Database Engineering**

11.1 Physical Data Model

The Physical data model visually represents the structure of the data as implemented by a relational database schema.

Topics

Topic	Detail	See also
Benefit	<p>In addition to providing a visual abstraction of the database structure, an important benefit of defining a Physical data model is that you can automatically derive the database schema from the model.</p> <p>This is possible due to the richness of meta-data captured by a Physical data model and its close mapping to aspects of the database schema, such as tables, columns, primary and foreign keys.</p>	Data Models [1625]
Example	<p>The following is an example of a Physical data model that could be used to automatically generate a database schema.</p> <p>Each table is represented by a UML Class; columns, including Primary and Foreign Keys, are modeled using UML attributes and operations.</p> <p>A Physical data model defined using a UML Profile and the Information Engineering notation</p>	
Notation	<p>The above model is defined using Enterprise Architect's <i>UML Profile for Data Modeling</i>; the relationship between the tables uses the Information Engineering notation.</p> <p>Information Engineering is one of three notations that Enterprise Architect supports to help Data Modelers distinguish cardinality in relationships.</p>	Data Modeling Notations [1982] Connectors Tab [732]
Set Default DBMS	<p>Prior to creating a Physical data model it is advisable to set the default DBMS. This pre-sets the default database when you create any new Table elements.</p> <p>If the DBMS is not set, Tables are created without a DBMS type and field typing cannot be allocated when creating new fields in the table.</p> <p>You can set the default DBMS type using either:</p> <ul style="list-style-type: none"> • The Settings Database Datatypes menu option or • The Code Generation Toolbar 	Add New Datatypes [1947] Set the Database Type [1938] Code Generation Toolbar [142]

Learn more

- [Create a Data Model Diagram](#)^[1933]
- [Tables and Columns](#)^[1935]
- [Data Types](#)^[1944]
- [Primary and Foreign Keys](#)^[1951]
- [Stored Procedures](#)^[1961]
- [Indexes](#)^[1963]
- [Triggers](#)^[1965]
- [Check Constraints](#)^[1966]
- [Views](#)^[1967]
- [Supported Databases](#)^[1970]
- [Oracle Packages](#)^[1970]
- [Forward and Reverse Engineer Databases](#)^[1931]
- [Transform Abstract Models to Physical Data Models Using MDA](#)^[1689]
- [Connect to Element Feature](#)^[944]

Learning Center topics

- **(Alt + F1) | Database Engineering**

11.1.1 Create a Data Model Diagram

Data Modeling diagrams are extended Class diagrams, and are used for modeling the structure of a relational database. These diagrams can include Tables, Views and Procedures, and can be set to use the connector notations *UML DDL Profile*, *IDEF1X* or *Information Engineering*.

Access **Project Browser package context menu | Add | Add Diagram**

Reference

Use the New Diagram dialog to create a Data Modeling diagram, as described below:

Field/Button	Action	See also
Name	This field defaults to the package name. If required, overwrite the default name with your preferred name.	
Select From	Click on Extended .	Add New Diagrams ^[724]
Diagram Types	Click on Data Modeling .	
OK	Click on this button.	Data Modeling Group ^[720]

Field/Button	Action	See also
	<p>The Diagram View displays the blank diagram, and the Data Modeling pages display in the Diagram Toolbox.</p> <p>Drag elements and connectors from the Toolbox onto your diagram, to create your data model.</p>	

Notes

- The new diagram connector notation defaults to **Information Engineering**; If you want to change this, display the Connectors tab of the diagram Properties dialog (**Diagram | Properties > Connectors**) and select the required option from the **Connector Notation** drop-down list

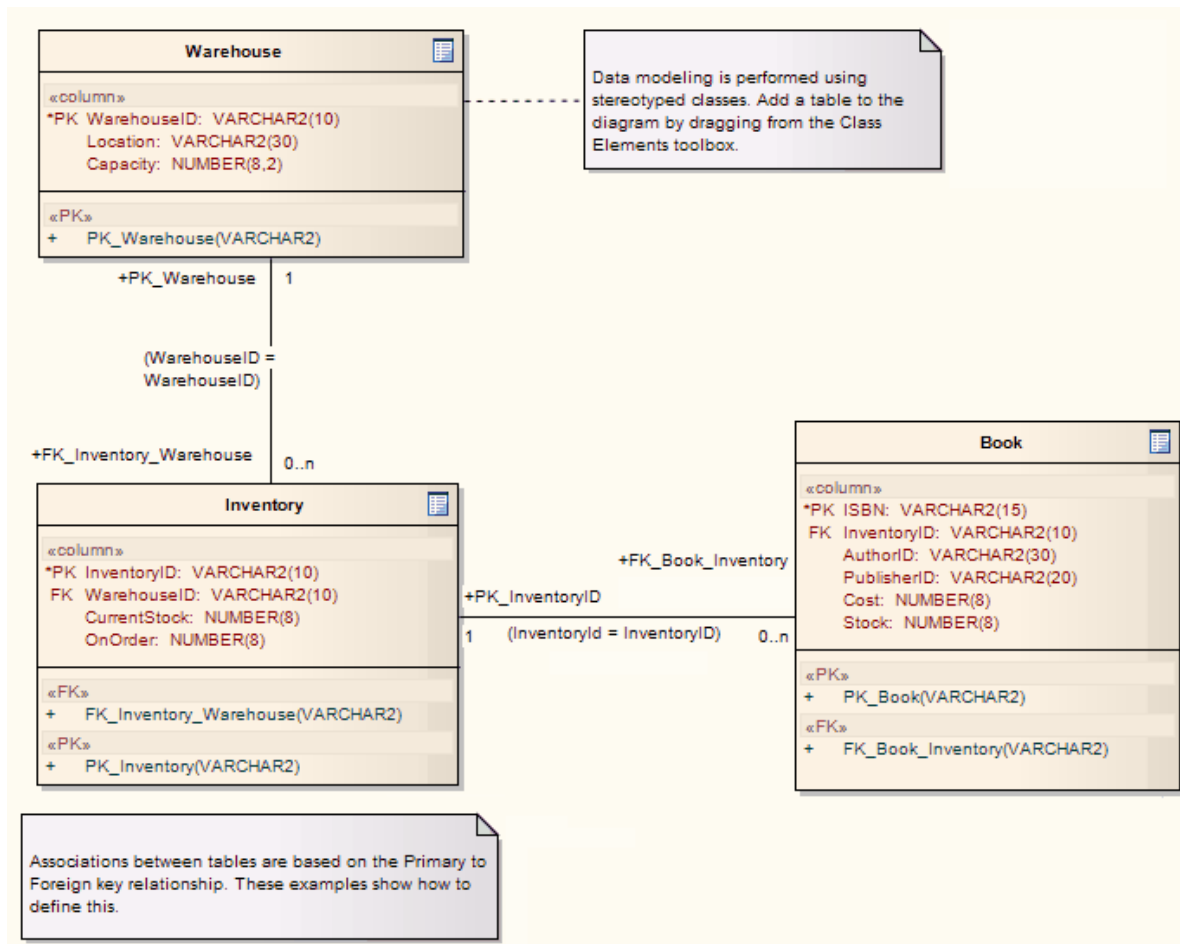
Learn more

- [Data Modeling Notations](#) ^[1982]
- [Connectors Tab](#) ^[732]
- [Example Data Model Diagram](#) ^[1934]

11.1.1.1 Example Data Model Diagram

An example of a *Data Model* diagram is provided below, showing three tables that are linked on *primary to foreign* key pairs with associated multiplicity.

Example



11.1.2 Tables and Columns

Tables and columns form the basis of Physical data models.

Topics

Topic	Detail	See also
Rows and Columns	<p>The basic structural element of a relational database is the <i>table</i>, which represents a set of records, or rows, that have the same structure.</p> <p>Each row contains one or more <i>columns</i>.</p> <p>Every individual item of data entered into a relational database is represented by a value in a column of a row in a table.</p> <p>Enterprise Architect's <i>UML Profile for Data Modeling</i> represents:</p> <ul style="list-style-type: none"> Database tables as Classes with a stereotype of <<table>> Columns by Attributes with a stereotype of <<column>> 	<p>Create a Table ¹⁹³⁶</p> <p>Create Columns ¹⁹⁴²</p>
Example	<p>A simple example of a Physical data model in Enterprise Architect is shown below. The example consists of two tables represented by</p>	<p>Add New Datatypes ¹⁹⁴⁷</p>

Topic	Detail	See also
	<p>UML Classes, named <i>SalesPerson</i> and <i>Customer</i>. The table stereotype is denoted by the icon in the top-right corner of each Class.</p> <p>The tables each define three columns using UML Attributes typed appropriately for the target DBMS, which happens to be MySQL.</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div data-bbox="435 548 707 723" style="border: 1px solid black; padding: 5px; width: 150px;"> <p style="text-align: center; margin: 0;">SalesPerson</p> <p style="margin: 0;">«column»</p> <p style="margin: 0;">Name :TEXT</p> <p style="margin: 0;">Department :TEXT</p> <p style="margin: 0;">* SalesPersonID :INTEGER</p> </div> <div data-bbox="839 548 1110 723" style="border: 1px solid black; padding: 5px; width: 150px;"> <p style="text-align: center; margin: 0;">Customer</p> <p style="margin: 0;">«column»</p> <p style="margin: 0;">Name :TEXT</p> <p style="margin: 0;">Address :TEXT</p> <p style="margin: 0;">* CustomerID :INTEGER</p> </div> </div> <p style="text-align: center;">A simple Data model consisting of two tables, represented by UML Classes</p>	
Default DBMS	<p>Prior to creating a tables in a Physical data model it is advisable to set the default DBMS. This pre-sets the default database when you create any new Table elements.</p> <p>If the DBMS is not set, Tables are created without a DBMS type and field typing cannot be allocated when creating new fields in the table.</p> <p>You can set the default DBMS type using either:</p> <ul style="list-style-type: none"> • The Settings Database Datatypes menu option or • The Code Generation Toolbar 	Add New Datatypes ^[1947]
Attribute/Field Mapping	<p>To help you map Class Attributes to Table fields, you can create connectors between specific attributes (features) in the Class element and the column Attributes in the Table element.</p>	Connect to Element Feature ^[944]
Model Driven Architecture	<p>Enterprise Architect can generate simple DDL scripts to create the tables in your model. You can also perform Model Driven Architecture (MDA) Transformations to DDL - Enterprise Architect provides a template specifically for DDL transformations.</p>	Model Transformation ^[1679] DDL Transformation ^[1689]
Primary and Foreign Keys	<p>Primary Key and Foreign Key information modeling is considered in later topics.</p>	Primary and Foreign Keys ^[1951]

11.1.2.1 Create a Table

Fundamental to data modeling is the creation of Tables within the model.

How to

To add a Table to your data model

Step	Action	See also
1	Create and/or open a Data Modeling diagram.	Create a Data Model Diagram [1933]
2	Click on the <i>Table</i> element in the Diagram Toolbox, then click on the diagram. The Table element is displayed on the diagram.	Data Modeling Group [720]
3	Use the Table Properties dialog to set the table name and other properties as required.	Working with Table Properties [1937]
4	Use the Quicklinker to add related Table elements.	The Quick Linker [791]

Learn more

- [Working with Table Properties](#) [1937]

Learning Center topics

- **(Alt + F1) | Database Engineering | Physical Data Model | Create Table**

11.1.2.1.1 Working with Table Properties

Once you have created a table, you must set its properties to define database-specific information that can be used to generate the schema in DDL. Table properties are set using the Properties dialog.

It is recommended that you set the DBMS you are using as a *default*, to automatically apply that DBMS definition to all tables you create. However, if you do not set a default, or want to change the value for specific tables for any reason, you can also identify the DBMS on the Table Properties dialog for each Table.

For MySQL and Oracle, some additional properties are defined using Tagged Values, such as the Table Owner and DBMS-specific options. For other DBMSs, additional table properties are not yet supported.

Topics

Topic	Link
<p>Set the Database type for a table - other than the table name, the most important property to set for a table is the database type, which defines:</p> <ul style="list-style-type: none"> • The list of data types that are available for defining columns, and • Which dialect of DDL is generated <p>You can either set the DBMS as a default for all tables, or within the Table Properties for each individual Table.</p>	<p>Add New Datatypes [1947]</p> <p>Set the Database Type [1938]</p>

Topic	Link
Set the Table Owner - Some relational databases enable you to assign a table to one of several owners within the database; Enterprise Architect enables you to model this using the <i>Owner</i> Tagged Value, so that in the generated DDL script the table name is prefixed by the owner name.	Set Table Owner/Schema ^[1939]
Example MySQL options - To make use of foreign keys in MySQL, you must declare the table type as InnoDB.	Set MySQL Options ^[1939]
Example Oracle options - To set additional Oracle table properties, use the table's Tagged Values.	Set Oracle Table Properties ^[1940]

Once the table properties are defined, you are ready to add columns.

Learn more

- [Create Columns](#) ^[1942]

11.1.2.1.1 *Set the Database Type*

The most important property to set for a table (after its name) is the database type. This defines the list of data types that are available for defining columns, and also declares which dialect of DDL is generated.

If this is not set, tables are created without a DBMS type and field typing cannot be allocated when creating new fields in the table.

You can set the database type to default to the required DBMS. However, if you have not set a default or want to change the value for specific tables, you can set the DBMS for each table individually.

Access **Table element | Element | Properties | General**

Reference

Field/Button	Action	See also
Database	Click on the drop-down button and select the required database type from the list.	
Apply	Save your changes.	
OK		

Learn more

- [Add New Datatypes](#)^[1947] (to set *default* DBMS)
- [Code Generation Toolbar](#)^[142]

Learning Center topics

- **(Alt + F1) | Database Engineering | Physical Data Model | DBMS Datatypes**

11.1.2.1.1.2 Set Table Owner/Schema

Some relational databases enable you to assign a table to one of several schemas (or *owners*) within the same database. Enterprise Architect enables you to model this information using the *Owner* Tagged Value. By setting this optional value, you can tailor the generated DDL script, causing the table name to be prefixed by its assigned owner name.

How to

To set the table owner Tagged Value

Step	Action	See also
1	Add a Tagged Value named <i>Owner</i> , to the Table element.	How to Add a Tagged Value ^[972]
2	Type the owner name for this table in the Value field of the Tagged Value. Changes are automatically saved.	

Notes

- For a PostgreSQL database, you should instead create a Tagged Value named OWNER TO with the corresponding value

11.1.2.1.1.3 Set MySQL Options

To make use of foreign keys in MySQL, you must declare the table type as InnoDB.

How to

To declare the table type as InnoDB

Step	Action	See also
1	Add a Tagged Value named Type to the table.	How to Add a Tagged Value ^[972]
2	Set the Value field to InnoDB .	

When you generate DDL for this table, the table type is included in the SQL script.

To allow for later versions of MySQL, additional table options that can be added in the same manner include:

Tag	Value (Example)
ENGINE	InnoDB
CHARACTER SET	latin1
CHARSET	latin1
COLLATE	latin1_german2_ci

11.1.2.1.1.4 Set Oracle Table Properties

To set additional Oracle table properties, you use the table's Tagged Values

How to

Step	Action	See also
1	Add one or more Tagged Values to the table, using the names provided in the Property/Tag column of the reference table below.	How to Add a Tagged Value 9721
2	Specify the appropriate value for each tag. Examples are provided in the Value column of the reference table below.	

The same properties can be added to indexes and constraints, by highlighting the index or constraint Operation and adding the appropriate Tagged Values.

Reference

Property/Tag	Value
BUFFER_POOL	DEFAULT
CACHE	NOCACHE
DBVERSION	9.0.111

Property/Tag	Value
FREELISTS	1
GRANT OWNER1	SELECT
GRANT OWNER2	DELETE, INSERT, SELECT, UPDATE
INITIAL	65536
INITRANS	1
LOGGING	LOGGING
MAXEXTENTS	2147483645
MAXTRANS	255
MINEXTENTS	1
MONITORING	MONITORING
OWNER	OWNER1
PARALLEL	NOPARALLEL
PCTFREE	10
PCTINCREASE	0
PCTUSED	0
SYNONYMS	PUBLIC:TABLE_PUB;OWNER2:TABLE_OWNER2
TABLESPACE	MY_TABLESPACE

Property/Tag	Value
TEMPORARY	YES

11.1.2.2 Create Columns

Columns are represented in the UML Data Modeling Profile as a stereotyped attribute; that is, an attribute with the *Column* stereotype.

How to

To create columns

Step	Action	See also									
1	Right-click on the required Table in a diagram, and select the Attributes context menu option. The Attributes dialog displays, showing the columns of the selected Table.										
2	In the Name field, type the column name.										
3	In the Data Type field, click on the drop-down arrow and select the data type (ensure the Table's target database is set first; this populates the Data Type drop-down list). Click on the Save button.	Set the Database Type ^[1938] Oracle Data Types ^[1949]									
4	Complete the required optional fields, listed in the reference table below.										
5	Click on the Column Properties button. The Database Columns Properties dialog displays. Supported column properties are identified below; it is assumed that you understand which data types these properties apply to:										
	<table border="1"> <thead> <tr> <th>Property</th> <th>DBMS</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td>Autonum</td> <td>Oracle MySQL SQL Server DB2 PostgreSQL Sybase ASA Sybase ASE</td> <td>If you require a sequence, select the Autonum property and set the value to True and, if necessary, define the start number and increment</td> </tr> <tr> <td>Zerofill</td> <td>MySQL</td> <td></td> </tr> </tbody> </table>	Property	DBMS	Notes	Autonum	Oracle MySQL SQL Server DB2 PostgreSQL Sybase ASA Sybase ASE	If you require a sequence, select the Autonum property and set the value to True and, if necessary, define the start number and increment	Zerofill	MySQL		
Property	DBMS	Notes									
Autonum	Oracle MySQL SQL Server DB2 PostgreSQL Sybase ASA Sybase ASE	If you require a sequence, select the Autonum property and set the value to True and, if necessary, define the start number and increment									
Zerofill	MySQL										

Step	Action	See also						
	<table border="1"> <tr> <td>Unsigned</td> <td>MySQL</td> <td></td> </tr> <tr> <td>LengthType</td> <td>Oracle</td> <td>Character semantics</td> </tr> </table>	Unsigned	MySQL		LengthType	Oracle	Character semantics	
Unsigned	MySQL							
LengthType	Oracle	Character semantics						
6	Click on the OK button to return to the Attributes dialog, and then click on the Save button.							
7	Click on the: <ul style="list-style-type: none"> • New button to define another column or • Close button to exit from the dialog 							

Reference

Field	Description	See also
Primary Key	Indicates whether the column is the primary key for this table.	Primary Key <small>[1952]</small>
Not Null	Indicates whether empty values are forbidden for this column.	
Unique	Indicates whether it is forbidden for any two values of this column to be identical. The unique <i>stereotype</i> applied to an index ensures that no two <i>combinations</i> of values across a <i>set</i> of columns can be identical.	
Initial	A value that can be used as a default value for this column, if required.	
Access	Determines the column's scope (Private , Protected or Public). The default value is Public .	
Alias	An alternative name for the field for display and documentation purposes.	
Notes	Additional information necessary to document the column. You can format the text using the Notes toolbar at the top of the field.	

Notes

- For MySQL, before creating columns you must first add ENUM and SET datatypes.

Learn more

- [Reorder Columns](#) ^[1944]
- [Data Types](#) ^[1944]
- [MySQL Data Types](#) ^[1949]
- [Options - Code Editors](#) ^[1895]

Learning Center topics



- **(Alt + F1) | Database Engineering | Physical Data Model | Add Column**

11.1.2.2.1 Reorder Columns

You can change the order in which columns are listed in a table, to define the order in which columns appear in the generated DDL that defines your database schema.

How to

To change the column order

Step	Action	See also
1	In the Columns dialog, highlight a column name in the Columns panel.	
2	Click on the: <ul style="list-style-type: none"> •  button to move the column up one position •  button to move the column down one position 	

11.1.3 Data Types

Each column that you define in your data model has a data type that is appropriate to the information being stored by that column.

Enterprise Architect contains definitions of the core datatypes for a number of standard DBMS products. However, data types vary from one DBMS product to another, and they also vary between versions of a given database product. Therefore Enterprise Architect provides you with tools to:

- Define new data types for a new version of your DBMS product
- Define data types for a new, non-standard database product
- Automatically convert data types from one defined DBMS product to another
- Define how the automatic conversion should proceed from or to a non-standard DBMS product.

Learn more

- [Supported Databases](#) ^[1970]

- [DBMS Product Conversion for a Package](#)^[1945]
- [Data Type Conversion For a Table](#)^[1946]
- [Add New Datatypes](#)^[1947]
- [Map Data Types Between DBMS Products](#)^[1950]

11.1.3.1 DBMS Product Conversion for a Package

The DBMS Package procedure or mapper enables you to automatically convert a package of database tables from one supported DBMS type to another supported DBMS type, as well as providing the ability to change the ownership of the tables.

You can also change the DBMS type for individual tables.

If one of the DBMS types is non-standard or otherwise not supported by Enterprise Architect, you should ensure that the mapping of datatypes from one DBMS type to the other has been defined.

Access [Project Browser Package context menu](#) | [Code Engineering](#) | [Reset DBMS Options](#)

Reference

On the Manage DBMS Options dialog map the DBMS data types of a package to the data types of another DBMS type, as indicated below:

Field/Button	Action	See also
Current DBMS	Click on the drop-down arrow and select the current DBMS.	
New DBMS	Click on the drop-down arrow and select the target DBMS.	
Convert DBMS Type	Select the checkbox.	
Process Child Packages	If there are child packages that also require changing, select the checkbox.	
OK	Click on this button to map all tables in the selected packages to the new DBMS.	

Also on the Manage DBMS Options dialog, you can change the owner of the selected table or all of the tables in a package, as indicated below:

Field/Button	Action	See also
New Owner	Type the name of the new table owner.	

Field/Button	Action	See also
Current Owner	Click on the drop-down arrow and select either: <ul style="list-style-type: none"> The current owner to change, or <All> to change the ownership of all tables in the package to the name you typed in the New Owner field 	
Change Table Owner	Select the checkbox.	
Process Child Packages	If there are child packages that also require changing, select the checkbox.	
OK	Click on this button to change the ownership of the selected Table, or all Tables in the selected packages that have the specified current owner.	

Learn more

- [Data Type Conversion For a Table](#) ^[1946]
- [Set Table Owner](#) ^[1939]
- [Diagram Properties](#) ^[725]
- [Map Data Types Between DBMS Products](#) ^[1950]

11.1.3.2 Data Type Conversion For a Table

Once a database schema has been set up on an Enterprise Architect diagram (either by importing through ODBC or manually setting up the tables), the DBMS can be changed to another type and the column datatypes are mapped accordingly.

You might follow this procedure if you have copied a small number of tables into the project from elsewhere, but if you have many tables you can also convert all of them at once within their parent package.

If one of the DBMS types is non-standard or otherwise not supported by Enterprise Architect, you should ensure that the mapping of datatypes from one DBMS type to the other has been defined.

How to

To map the DBMS type of a table to another DBMS type:

Step	Action	See also
1	Double-click on the Table element in a diagram. The Table Properties dialog displays, with the Database field showing the current DBMS for this table.	
2	To map the datatypes to another DBMS, click on the Database drop-down arrow	

Step	Action	See also
	and select the target DBMS.	
3	Click on the Apply button. The datatypes are converted to match those of the new DBMS, and these are reflected in any DDL generated from this table.	

Learn more

- [DBMS Product Conversion for a Package](#)^[1945]
- [Map Data Types Between DBMS Products](#)^[1950]

11.1.3.3 Add New Datatypes

Using Enterprise Architect's Database Datatypes dialog, you can add to the set of data types that are available for a particular DBMS.

Access **Settings | Database Datatypes**

Use to

- Identify the DBMS in use and, if required, set this as the model default
- Include any new datatypes that are supported by later versions of the DBMS and not yet included with Enterprise Architect
- Remove any datatypes that are no longer relevant
- Add a new DBMS product and its built-in data types if, for example, you want to create a physical data model for a DBMS product that is not yet supported natively by Enterprise Architect
- Change the DBMS in use and begin to map the datatypes to the new DBMS

Reference

Use the Database Datatypes dialog to add a new datatype to a DBMS product, as described below:

Field/Button	Action	See also
Product Name	Click on the drop-down arrow and select an existing DBMS.	
Add Product	If your DBMS is not listed, click on this button to add it. An Input prompt displays, in which you type the DBMS name; click on the OK button to add the name to the drop-down list.	
Set as Default	Select the checkbox to set the selected DBMS as the default for your database engineering and modeling.	Physical Data Model ^[1932]

Field/Button	Action	See also
	<p>Once you set the default database, when you create any new Table element the Database type is automatically pre-set to this default.</p> <p>You can also set the default database type through the Code Generation Toolbar.</p>	Code Generation Toolbar ^[142]
Datatype	Type a name for the data type.	
Size	Select the appropriate radio button and, if appropriate, specify the default and maximum values.	
Common Type	Click on the drop-down arrow and select a common type to which your data type applies.	
Save	Click on the button to create your data type and add it to the Defined Datatypes for Databases list.	
Delete	Select a data type in the Defined Datatypes for Databases list and click on this button to remove the data type.	
Datatype Map	If you have changed the DBMS or technology for which you have defined the data types from or to an unsupported DBMS type, click on this button to define how to automatically remap the data types to your new DBMS or technology.	Map Data Types Between DBMS Products ^[1950]

You can transport these data types between models as Enterprise Architect Reference Data.

Learn more

- [Add MySQL Data Types](#) ^[1949]
- [Customize Oracle Data Types](#) ^[1949]
- [Export Data Types as Reference Data](#) ^[306]
- [Import Data Types as Reference Data](#) ^[308]

Learning Center topics

- **(Alt + F1) | Database Engineering | Physical Data Model | Data Type Map**

11.1.3.3.1 MySQL Data Types

MySQL supports the **ENUM** and **SET** data types, which must be added to your Enterprise Architect model before you can use them as the types for columns.

Access [Settings | Database Datatypes](#)

How to

To add the ENUM and SET data types for MySQL

Step	Action	See also
1	Select the Database Datatypes menu option. The Database Datatypes dialog displays.	
2	In the Product Name field select MySQL .	
3	Add the data types ENUM and SET .	

When using these data types later in a Column's **Initial** field, type the values as a comma-separated list, in the format:

```
(one,two,three)
```

If one value is the default, use the format:

```
(one,two,three) default three
```

11.1.3.3.2 Oracle Data Types

The Oracle data types **NUMBER** and **VARCHAR** have additional properties that you can model.

Reference

Data Type	Detail	See also
NUMBER	The NUMBER data type requires precision and scale properties. The Precision and Scale fields are displayed on the Attributes dialog when the data type is set to NUMBER ; if you enter information into these fields, it is displayed on your diagrams. For example: create NUMBER by setting Precision = 0 and Scale = 0 create NUMBER(8) by setting Precision = 8 and Scale = 0 create NUMBER(8,2) by setting Precision = 8 and Scale = 2	
VARCHAR	Oracle VARCHAR2(15 CHAR) and VARCHAR2(50 BYTE) data types can be	

Data Type	Detail	See also
	created by adding the Tagged Value LengthType with the value CHAR or BYTE .	

Learn more

- [How to Add a Tagged Value](#)^[972]

11.1.3.4 Map Data Types Between DBMS Products

One of the advantages of using visual physical data models to hide implementation details is that you can more easily change the target technology or platform when required.

After reverse engineering your database into a physical data model, you must remap the data types before generating the schema for the new DBMS product. Enterprise Architect provides a set of default mappings for standard, supported DBMS products, to help you automate the conversion process.

However, you might want to customize the default mappings to suit your specific project requirements, or where the mapping of one data type to another is not currently defined. For example, in your migration from one DBMS platform to another, one of the platforms might be non-standard or otherwise not supported by Enterprise Architect.

Access **Settings | Database Datatypes: Datatype Map**

Use to

- Streamline conversion of one implementation-specific database product to another
- Map data types from one DBMS product to another
- Maximize portability of your database designs
- Customize default mappings

Reference

On the Database Datatypes Mapping dialog, map the data types between databases as indicated below:

Field/Button	Action	See also
From Product Name	Click on the drop-down arrow and select the DBMS product to map data types from.	
Defined Datatypes for Databases	Displays all the defined data types for the product and, where appropriate, their sizes and values. Click on the data type to map - this must have a defined size unit and value. The Datatype and Common Type fields under the From Product Name field display this data type.	

Field/Button	Action	See also
To Product Name	Click on the drop-down arrow and select the DBMS product to map data types to. The Datatype and Common Type fields under this field display the values corresponding to those in the fields for the 'From' product.	
Size	Click on the radio button for the appropriate size unit and type the default values in the corresponding data fields.	
Save	Click on this button to save the mapping.	

Repeat this process for all the data types to map.

Once you are satisfied with the data type mappings, you can convert either individual tables or an entire package of tables to the new target DBMS product.

Learn more

- [DBMS Product Conversion for a Package](#)^[1945]
- [Data Type Conversion For a Table](#)^[1946]

Learning Center topics

- **(Alt + F1) | Database Engineering | Physical Data Model | DBMS Migration**

11.1.4 Database Keys

Two types of key are used to access tables in a relational database: *Primary Keys* and *Foreign Keys*.

- A Primary Key uniquely identifies a record in a table
- A Foreign Key accesses data in some other related table via its Primary Key

Topics

Topic	Detail	See also
General Usage	In Enterprise Architect, Primary Keys and Foreign Keys are represented by stereotyped UML attributes and operations.	
Example	The following example shows a data model that includes both Primary Key and Foreign Key columns.	

Topic	Detail	See also
	<p>A Physical data model defined using a UML Profile that includes Primary and Foreign Keys</p>	
Primary Key	In each table, one of the columns is designated as the Primary Key, indicated by the <i>PK</i> label beside the column, and a corresponding Primary Key Operation.	Primary Key ^[1952]
Foreign Keys	Foreign Keys are shown denoted by the <i>FK</i> labels.	Foreign Key ^[1956]
Association Notation	The Association that defines a Foreign Key relationship between <i>SalesPerson</i> and <i>Customer</i> uses the Information Engineering notation; this is one of three notations that Enterprise Architect supports to help Data Modelers distinguish cardinality in relationships. Enterprise Architect also supports: <ul style="list-style-type: none"> • IDEF1X notation • UML notation 	Data Modeling Notations ^[1982]

Learn more

- [Primary Keys](#)^[1952]
- [Foreign Keys](#)^[1956]

11.1.4.1 Primary Key

A Primary Key uniquely identifies a record in a table and can consist of one or more columns.

Topics

Topic	Detail	See also
Types of Primary Key	In Enterprise Architect, a: <ul style="list-style-type: none"> • Simple Primary Key (single column) is defined as the attribute of a stereotyped operation • Complex Primary Key (several columns) is defined as the stereotyped operation itself 	Create a Primary Key ^[1953]

Topic	Detail	See also
	For example, a column <i>LastName</i> might not be unique within a table, so a Primary Key is created from the <i>LastName</i> , <i>FirstName</i> and <i>DateOfBirth</i> columns.	

Learn more

- [Define a Primary Key Name Template](#)^[1954]
- [Foreign Keys](#)^[1956]

Learning Center topics

- **(Alt + F1) | Database Engineering | Physical Data Model | Add Primary Key**

11.1.4.1.1 Create a Primary Key

How to

To define a simple Primary Key consisting of a single column

Step	Action	See also
1	Right-click on the table in a diagram and select the Attributes context menu option. The Attributes dialog displays.	
2	Select the column that makes up the Primary Key.	
3	Select the Primary Key checkbox and click on the Save button. A stereotyped operation is automatically created; it is this operation that defines the Primary Key for the table. To remove a Primary Key, simply delete this operation.	

To define a complex Primary Key, consisting of more than one column

Step	Action	See also
1	Follow the steps above to create a Simple Primary Key; it doesn't matter which column you choose.	

Step	Action	See also
2	Right-click on the table in a diagram and select the Operations context menu option. The Operations dialog displays.	
3	Select the Primary Key operation (its name begins with PK_) and then click on the Column tab.	
4	Click on the New button to add a column to the Primary Key, select a column from the Column Name list box, and then click on the Save button.	
5	Click on the Hand buttons (up and down arrow) to change the order of columns in the Primary Key, if necessary.	

Learn more

- [Creating Non-Clustered Primary Keys for SQL Server](#)^[1955]

11.1.4.1.2 Define Primary Key Name Template

Enterprise Architect enables you to customize the way Primary Keys are named by default. This is helpful if you have specific modeling conventions that stipulate how Primary Keys are documented.

By changing the **Primary Key Name Template**, you adjust the name that Enterprise Architect assigns to the UML Operation that represents the Primary Key.

Access [Tools](#) | [Options](#) | [Source Code Engineering](#) | [Code Editors](#): **DDL Name Templates**

How to

To define the name template for a Primary Key

Step	Action	See also
1	Click on the DDL Name Templates button. The DDL Name Template dialog displays, showing the default name templates.	
2	Edit or replace the template in the Primary Key Name Template field. For example, if you want to display the Primary Key description as <i>PK_tablename_columnname</i> then change the Primary Key Name Template field to <i>PK_%tablename%_%columnname%</i> .	

Step	Action	See also
3	Click on the Save button.	

11.1.4.1.3 SQL Server Non Clustered Keys

When you create a primary key in SQL Server, it is created with a 'clustered index' by default. Therefore when you model a primary key for SQL Server in an Enterprise Architect data model, the corresponding DDL creates a clustered index for that primary key by default.

Clustered indexes provide improved performance for accessing the column(s) involved, but only one clustered index is allowed per table.

In some situations, you might be more interested in the performance of columns other than the Primary Key, and therefore must change the default assignment of the clustered index. Enterprise Architect allows you to model this.

How to

To define a Primary Key as non-clustered for a SQL Server table

Step	Action	See also
1	Right-click on the table in a diagram and select the Operations context menu option. The Table Operations dialog displays.	
2	Highlight the Primary Key Operation and click on the Extended Properties button. The Database Operation Properties dialog displays.	
3	Select the SQL Server Non Clustered Primary Key checkbox.	
4	Click on the Save & Close button.	

Subsequently, you can model an index for that table and define it as clustered instead.

Learn more

- [Indexes](#)^[1963]

11.1.4.2 Foreign Key

A Foreign Key accesses data in some other related table via its Primary Key.

Topics

Topic	Detail	See also
The nature of Foreign Keys	A Foreign Key consists of a collection of columns (UML Attributes) that together have some operational meaning (they enforce a relationship to the Primary Key in the other table).	
Modeling	Foreign keys are modeled in Enterprise Architect as operations with the stereotype FK . The operation's parameters become the columns involved in the key.	
Necessity of Use	It is not necessary to define a Foreign Key just to access another table through its Primary Key. Foreign Keys are a feature of some database management systems, providing 'extras' such as referential integrity checking that prevents the deletion of a record if its Primary Key value exists in some other table's Foreign Key. The same thing can be achieved programmatically.	

Learn more

- [Create a Foreign Key](#)^[1958]
- [Composite Foreign Key](#)^[1959]
- [Define a Foreign Key Name Template](#)^[1960]

Learning Center topics

- **(Alt + F1) | Database Engineering | Physical Data Model | Add Foreign Key**

11.1.4.2.1 Create a Foreign Key

How to

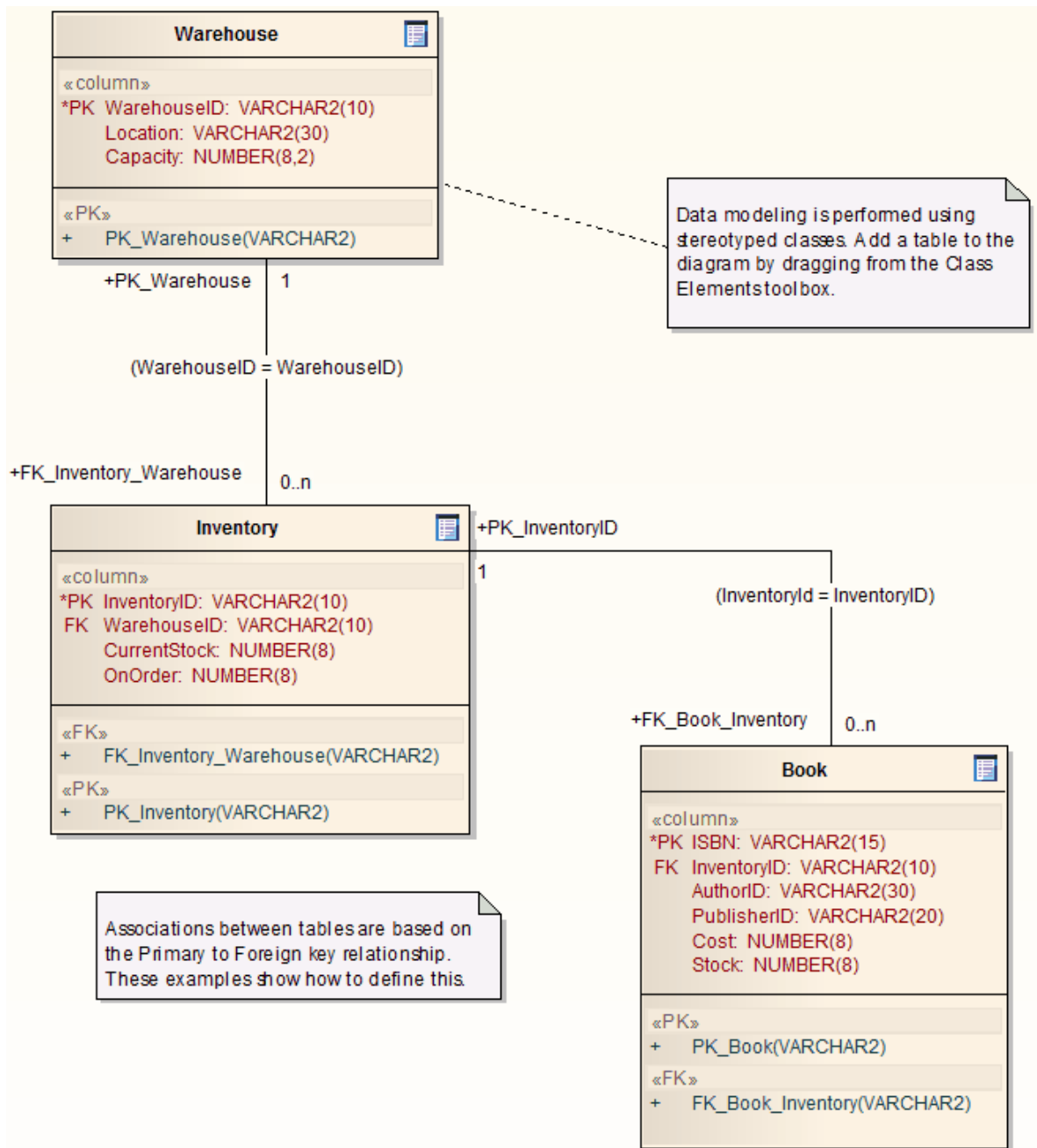
To create a Foreign Key

Step	Action	See also
1	Locate the required Tables in a diagram. Both tables must have defined database types .	Set The Database Type ^[1938]

Step	Action	See also
2	Select an <i>Associate</i> connector in the Class Relationships page of the Toolbox.	
3	Click on the Table to contain the Foreign Key (source) and draw the connector to the other Table (target).	
4	Right-click on the connector and select the Foreign Keys context menu option. The Foreign Key Constraint dialog displays.	
5	The default foreign key name is set by the Foreign Key Name Template. To change the name to something other than the default provided by the template, select the Override Template checkbox and edit the foreign key name.	Define Foreign Key Name Template <small>[1960]</small>
6	In the Source: panel and the Target: panel, click on the name of each of the two columns involved in the Foreign Key relationship.	
7	From the On Delete and/or On Update combo boxes, select the appropriate referential integrity constraint.	
8	Click on the Apply or OK buttons to automatically generate the Foreign Key operations.	

Example

The example below shows how a Foreign Key looks in a diagram:



Notes

- For MySQL databases, Foreign Keys are supported for InnoDB tables only.

InnoDB is the default MySQL storage engine as of version 5.5; if you are modeling a MySQL database that is earlier than version 5.5 and you want to use Foreign Keys, you must ensure the **Engine** Tagged Value is set to **InnoDB**



Learn more

- [Set MySQL Options](#) [1939]

11.1.4.2.2 Composite Foreign Key

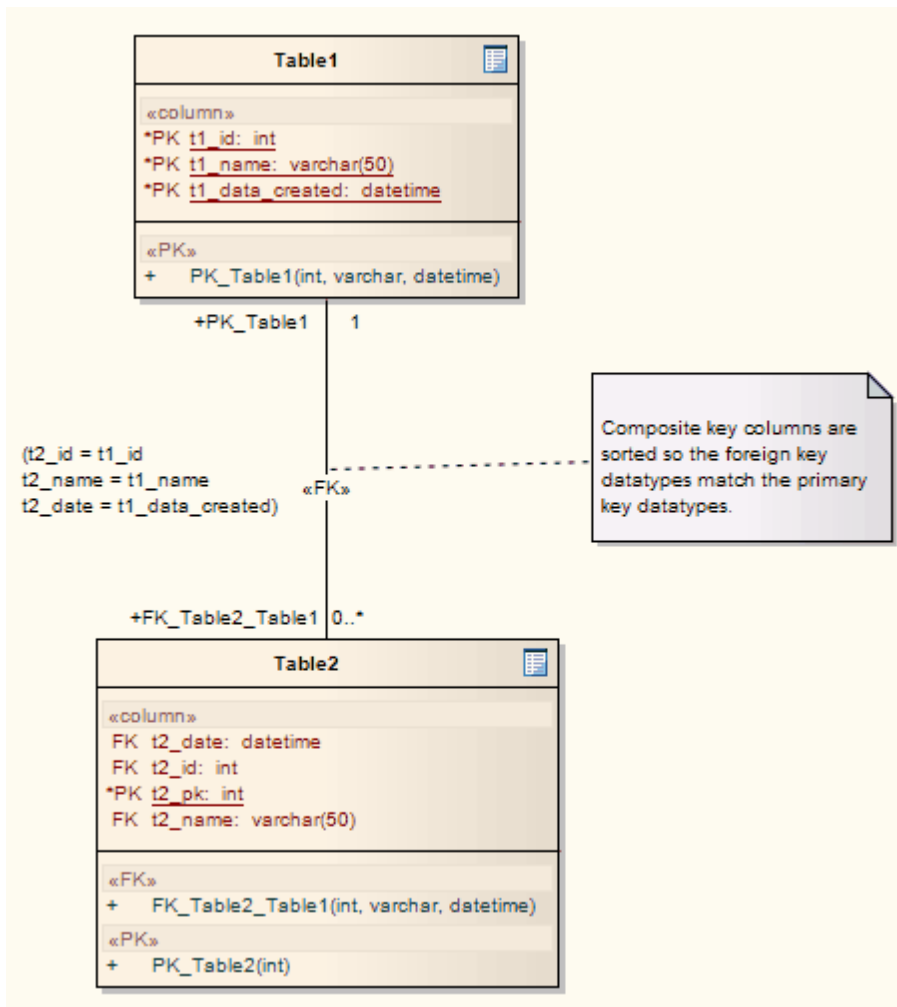
How to

To create a composite Foreign Key

Step	Action	See also
1	Locate the required Tables in a diagram. Both tables must have defined database types .	Set The Database Type ^[1938]
2	Select an <i>Associate</i> connector in the Class Relationships page of the Toolbox.	
3	Click on the Table to contain the Foreign Key (source) and draw the connector to the other Table (target).	
4	Right-click on the connector and select the Foreign Keys context menu option. The Foreign Key Constraint dialog displays.	
5	The default foreign key name is set by the Foreign Key Name Template. To change the name to something other than the default provided by the template, select the Override Template checkbox and edit the foreign key name.	Define Foreign Key Name Template ^[1960]
6	In the Source: panel and the Target: panel, click on the names of the required columns in each panel.	
7	Click on the OK button. The Foreign Key columns are sorted according to data type to match the data types of the targeted composite Primary Key.	
8	If required, you can change the order of the key columns by clicking on the  and  buttons.	

Example

The example below shows how a Composite Foreign Key looks in a diagram:



11.1.4.2.3 Define Foreign Key Name Template

Enterprise Architect enables you to customize the way Foreign Key relationships are named by default. This is helpful if you have specific modeling conventions that stipulate how Foreign Keys are documented.

By changing the **Foreign Key Name Template**, you adjust the name that Enterprise Architect assigns to the Association connector and the Operation that define the Foreign Key relationship.

Access [Tools | Options | Source Code Engineering | Code Editors: DDL Name Template](#)

How to

To define the name template for a Foreign Key

Step	Action	See also
1	Click on the DDL Name Template button. The DDL Name Template dialog displays, showing the default name templates.	

Step	Action	See also
2	<p>Edit or replace the name template in the Foreign Key Name Template field.</p> <p>For example, if you want to display the Foreign Key description as:</p> <p><i>FK_foreigntablename_FKcolumnname_primarytablename_PKcolumnname</i></p> <p>then change the Foreign Key Name Template field to:</p> <p><i>FK_%foreigntablename%_%fkcolumnname%_%primarytablename%_%pkcolumnname%</i></p>	
3	Click on the Save button.	

11.1.5 Stored Procedures

Stored Procedures are subroutines for processing table data on the DBMS server. These routines can be called from an external application.

Topics

Topic	Detail	See also
What is a Stored Procedure?	<p>A stored procedure is a group of SQL statements that perform a specific task; they are used to group a set of operations or queries that can be executed on a database server.</p> <p>Stored Procedures are often used where there is a need to reduce network usage between clients and the DBMS servers.</p> <p>Enterprise Architect models stored procedures using stereotyped UML Classes and Operations. These can be generated with a schema for loading onto the DBMS server.</p>	<p>Import Database Schema ^[1974]</p> <p>Create a Stored Procedure ^[1967]</p> <p>DBMS Product Conversion for a Package ^[1945]</p>

Learning Center topics

- (Alt + F1) | Database Engineering | Physical Data Model | Stored Procedures

11.1.5.1 Create a Stored Procedure

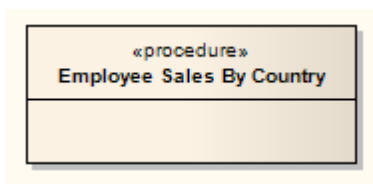
How to

To create a stored procedure as a UML Class, follow the steps below.

Step	Action	See also
1	Open the required diagram.	

Step	Action	See also
2	From the Data Modeling page of the Toolbox (More tools Data Modeling) drag the <i>Procedure</i> icon onto the diagram.	
3	If the Properties dialog does not automatically display, double-click on the element.	
4	In the Database field click on the drop-down arrow and select the target DBMS to model. The field displays the default database if it has already been set.	Set the Database Type ^[1938]
5	In the Procedure definition field, type the entire procedure text.	Languages supported ^[1800] Options - Code Editors ^[1895]
6	Click on the OK button.	
7	To rename the stored procedure, select the element, press (F2) and type the new name.	

Example



Notes

- Stored Procedures can also be defined under a single Class with the stereotype set to <<Stored Procedures>>, where each Stored Procedure is created as an operation under that Class; the scripting of these Stored Procedures can be viewed in the Operations dialog under **Behavior**

Learn more

- [Generate DDL](#)^[1978]
- [Import Database Schema](#)^[1974]

11.1.6 Indexes

Topics

Topic	Detail	See also
What is an Index?	<p>An index is a sorted look-up for a table; when it is known in advance that a table must be sorted in a specific order, it is usually worth the small processing overhead to always maintain a sorted look-up list rather than sort the table every time it is required.</p> <p>In Enterprise Architect, an index is modeled as a stereotyped operation; on generating DDL, the necessary instructions for generating indexes are written to the DDL output.</p>	

Learn more

- [Create an Index](#)¹⁹⁶³

Learning Center topics

- **(Alt + F1) | Database Engineering | Physical Data Model | Add Index**

11.1.6.1 Create an Index

How to

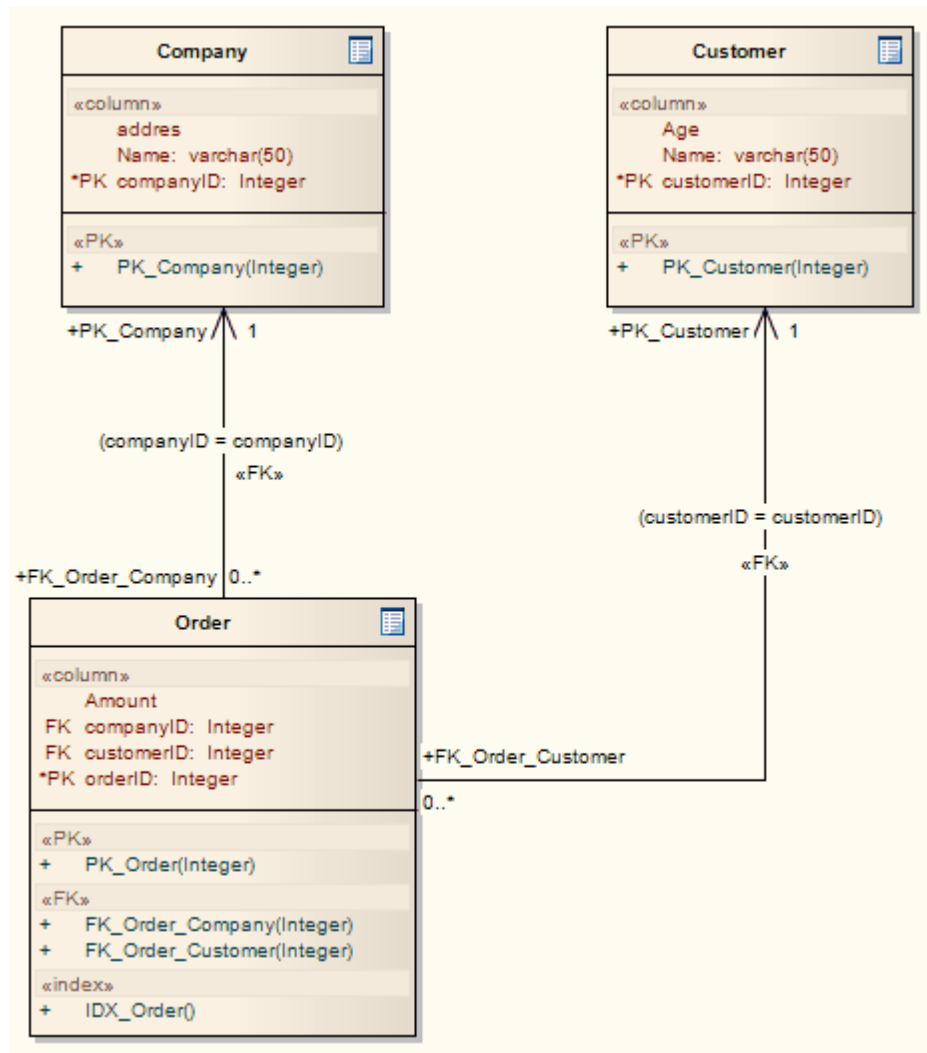
To create an Index

Step	Action	See also
1	Ensure that the column(s) to be used in the index have already been defined in the table.	
2	Right-click on the required table either in a diagram or in the Project Browser, and select the Operations context menu option. The Operations dialog displays.	
3	Add an operation with a name such as <i>IDX_CustomerID</i> . The <i>IDX_</i> prefix is optional, but helps to identify the operation.	
4	In the Stereotype field for the operation, select index (check and unique are also supported).	
5	Click on the Column tab.	

Step	Action	See also
6	Select the required columns from the Columns drop-down list in the required order.	
7	Click the Save button to save changes.	

Example

In the example below, the *Order* element shows how an index looks in a diagram:



Notes

- The **unique** characteristic applied to a single column ensures that no two *data values* in the column can be identical; the **unique** stereotype applied to an index ensures that no two *combinations* of values

across a set of columns can be identical

Learn more

- [Define Columns](#)^[1942]

11.1.7 Triggers

Topics

Topic	Detail	See also
What is a Trigger?	<p>A trigger is an operation automatically executed as a result of the modification of data in the database, and usually ensures consistent behavior of the database.</p> <p>For example, a trigger might be used to define validations that must be performed every time a value is modified, or might perform deletions in a secondary table when a record in the primary table is deleted.</p> <p>In Enterprise Architect, a trigger is modeled as a stereotyped operation.</p>	

Learn more

- [Create a Trigger](#)^[1965]

Learning Center topics

- [\(Alt + F1\) | Database Engineering | Physical Data Model | Create Trigger](#)

11.1.7.1 Create a Trigger

How to

To create a Trigger

Step	Action	See also
1	Locate the required table in either a diagram or the Project Browser.	
2	Right-click on the table and select the Operations context menu option. The Operations dialog displays.	
3	Add an operation with a name such as <i>TRG_OnCustomerUpdate</i> . The <i>TRG_</i> prefix is optional, but helps to identify the operation.	

Step	Action	See also
4	In the Stereotype field select trigger .	
5	Click the Save button.	
6	Select the trigger operation, then the Behavior tab.	
7	In the Initial Code field, enter the entire trigger code including the <code>CREATE_TRIGGER</code> statement.	Languages supported ^[1800] Options - Code Editors ^[1895]
8	Click on the Save button.	

11.1.8 Check Constraints

Topics

Topic	Detail	See also
What is a Check Constraint?	A <i>Check Constraint</i> enforces domain integrity by limiting the values that are accepted by a column.	

Learn more

- [Create a Check Constraint](#)^[1966]

Learning Center topics

- **(Alt + F1) | Database Engineering | Physical Data Model | Check Constraint**

11.1.8.1 Create a Check Constraint

How to

To create a Constraint

Step	Action	See also
1	Locate the required table in either a diagram or the Project Browser.	

Step	Action	See also
2	Right-click on the table and select the Operations context menu option. The Operations dialog displays.	
3	Add an operation with a name such as <i>CHK_ColumnName</i> . The <i>CHK_</i> prefix is optional, but helps to identify the operation.	
4	In the Stereotype field select check .	
5	Click on the Save button.	
6	Select the check constraint operation, then the Behavior tab.	
7	In the Initial Code field, enter the entire check constraint clause (for example, col1 < 1000).	
8	Click on the Save button.	

11.1.9 Views

Topics

Topic	Detail	See also
What is a Database View?	<p>A Database View defines a subset of the database, aggregated into a dynamic, 'virtual' table.</p> <p>The information presented by a Database View is not physically replicated, rather it is automatically derived based on the query that defines the view.</p> <p>Enterprise Architect models Database Views as stereotyped UML Classes.</p>	

Learn more

- [Create a View](#)^[1968]

Learning Center topics

- (Alt + F1) | Database Engineering | Physical Data Model | Create View

11.1.9.1 Create a View

How to

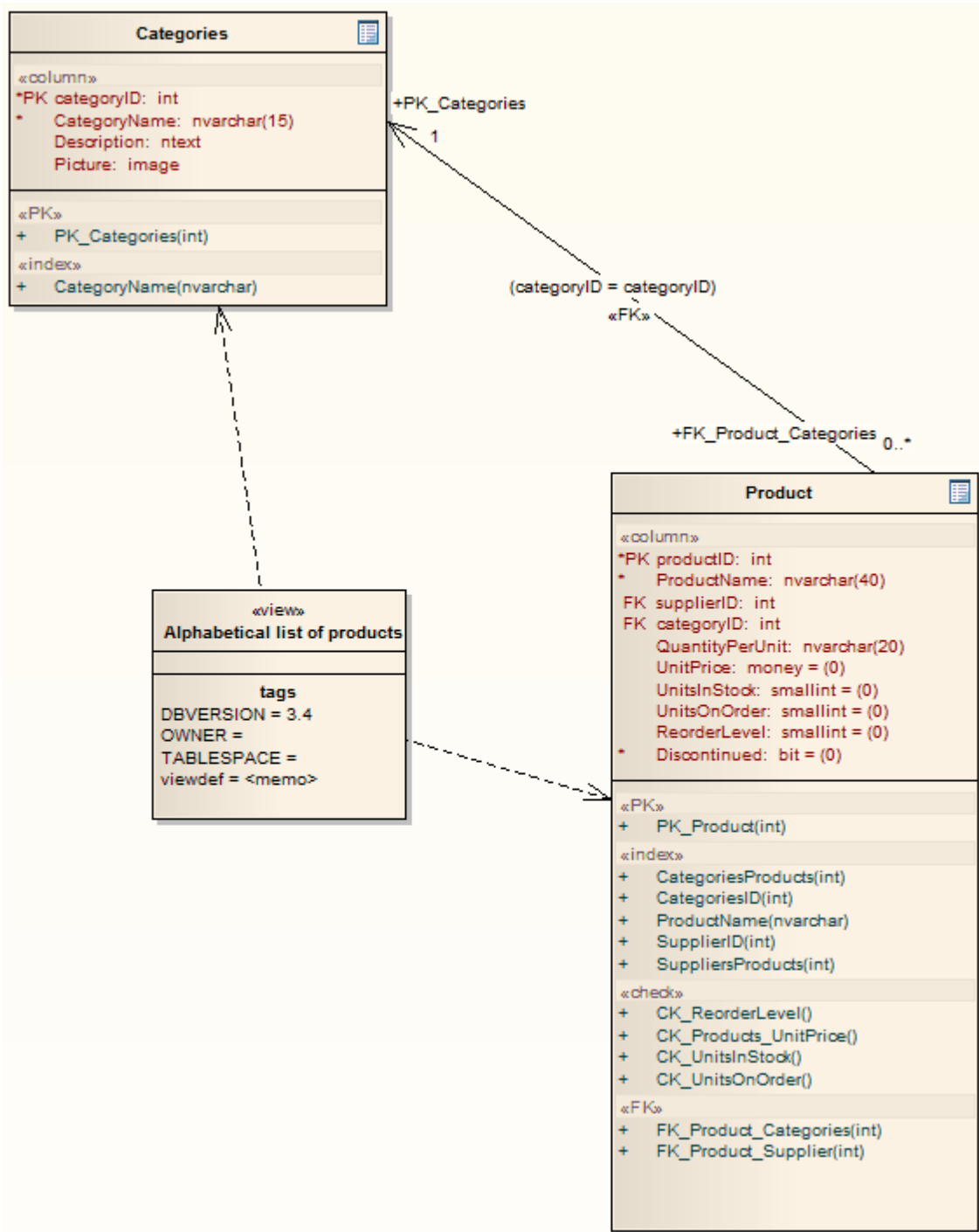
To create a database View

Step	Action	See also
1	On the Data Modeling page of the Toolbox (More tools Data Modeling), drag the <i>View</i> icon onto your Data Modeling diagram.	
2	If the View Properties dialog does not immediately display, double-click on the element.	
3	From the Database drop-down list, select the target DBMS to model. The default database displays if it has already been set.	Set the Database Type ^[1938]
4	Click on the OK button.	
5	To rename the View, select the element, press (F2) and type the new name.	

To create the View definition, follow the steps below

Step	Action	See also
1	Create a Dependency connector from the View to the table or tables on which the View depends.	
2	Double-click on the View to display the Properties dialog. The tables are now listed in the Dependencies field.	
3	In the View definition field, type the full view definition. The code editor provides intellisense for basic SQL keywords and functions.	Options - Code Editors ^[1895] Intellisense ^[1812]
4	Click on the OK button to save your definition.	

Example



Notes

- The View definition and certain other parameters are held as Tagged Values; the View definition is held in the *viewdef* memo Tagged Value

You can select and view the *viewdef* Tagged Value in the Tagged Values window, and include it in RTF

reports by inserting the **valueOf(viewdef)** field in the *Package::Package Element* or *Package::Element* sections.

Learn more

- [Tagged Values](#)^[969]
- [valueOf\(viewdef\)](#)^[2314]

11.1.10 Oracle Packages

How to

To create an Oracle package

Step	Action	See also
1	Add a Class element to your data model.	
2	Open the Properties dialog for the element and, in the Stereotype field, type the value Package .	
3	For the package specification, create an Operation with the name <i>Specification</i> and with no return type.	
4	Open the Properties dialog for the <i>Specification Operation</i> and, on the Behavior tab, type the entire package specification into the Initial Code field.	
5	For the package body, create an Operation with the name <i>Body</i> and with no return type.	
6	Open the Properties dialog for the <i>Body Operation</i> and, on the Behavior tab, type the entire package body into the Initial Code field.	

Learn more

- [How to Use the Initial Code Field](#)^[906]

11.1.11 Supported Databases

Topics

Topic	Detail	See also
Supported DBMS	Enterprise Architect supports modeling and reverse engineering of	

Topic	Detail	See also
Products	database schema for these DBMS products: <ul style="list-style-type: none"> • DB2 • Firebird/InterBase* • Informix • Ingres • MS Access 97, 2000, 2003 • Access 2007 • MS SQL Server 2000, 2005, 2008 • MySQL • SQLite • Oracle 9i, 10g and 11g • PostgreSQL • Sybase Adaptive Server Anywhere (Sybase ASA) • Sybase Adaptive Server Enterprise (Sybase ASE) 	

Notes

- * Firebird 1.5 database tables can be modeled and generated as InterBase tables; Firebird tables can be imported, but are treated as InterBase tables
- You can download data types for SQL Server 2005, SQL Server 2008, Informix and MS Access 2007 from the Resources page of the Sparx Systems web site

Learn more

- [Online Resources](#)

11.1.12 Default Constraints

Default Constraints only apply to SQL Server, and are supported to maintain compatibility with earlier versions.

Access **Table context menu | Operations**

Use to

- Create a default constraint

How to

To create a Default Constraint using the Operations Properties dialog

Step	Action
1	In the Name field, type the new operation name DF_<ColumnName> . The DF_ is optional, but helps to identify the operation.
2	In the Stereotype field type or select the value default .
3	Click on the Save button.
4	In the Operations list, click on the default operation name, then click Column in the left panel.
5	Select the appropriate column from the Column Name drop down list.
6	Click on Behavior in the left-hand panel.
7	In the Initial Code field, type the default value.
8	Click on the Save button.

11.2 Import Database Schema

Enterprise Architect provides facilities for reverse engineering DBMS schemas, enabling you to analyze and remodel legacy database systems and then export them to the existing or an alternative DBMS.

By connecting to a live database via ODBC, you can import the database schema into a standard UML model. Subsequent imports enable you to update your model from the live database.

Enterprise Architect supports importing database tables, stored procedures and views from an ODBC data source. Tables are imported as stereotyped Classes with suitable data definitions for the source DBMS.

How to

To import database schema and objects

Step	Action	See also
1	<p>Select a package in the Project Browser, into which to import the database schema.</p> <p>Create a data modeling diagram in this package.</p>	<p>Create a Data Model Diagram ^[1933]</p>
2	<p>To import, either:</p> <ul style="list-style-type: none"> • Right-click on the package and select the Code Engineering Import DB Schema from ODBC context menu option, or • Right-click on the diagram and select the Import DB schema from ODBC context menu option, or • Select the Tools Database Engineering Import DB schema from ODBC menu option <p>The Import DB Schema from ODBC Source dialog displays.</p>	
3	<p>In the Database field, click on the (...) (Browse) button and select a suitable ODBC data source from the ODBC dialog (ODBC must be installed and configured on your machine for this to work correctly).</p> <p>When you have selected the data source, the Database field shows the DBMS, the database server ID and the database name, separated by full stops; that is:</p> <p><i>dbms.dbserver.database.</i></p>	<p>Select a Suitable Data Source ^[1975]</p> <p>See <i>Notes</i> below, concerning 32-bit ODBC drivers.</p>
4	<p>You can filter objects to be retrieved from the database by schema or owner.</p> <p>In the Schema/Owner field, type the schema/owner as a single entry, or as a comma-separated list.</p> <p>The filter is useful for databases that support multiple schemas or owners, such as SQL Server 2005/2008, Oracle, PostgreSQL and DB2 Express.</p>	

Step	Action	See also
	<p>By default, when importing database objects from Oracle, Enterprise Architect inserts the Oracle user name in this field to avoid unnecessary retrieval of large numbers of objects not owned by that user.</p>	
5	<p>Select the appropriate checkboxes to import system tables and views, user views, triggers and/or Oracle packages.</p>	<p>Tables and Columns^[1935]</p> <p>Views^[1967]</p> <p>Triggers^[1965]</p>
6	<p>Select the appropriate checkboxes to import:</p> <ul style="list-style-type: none"> • Stored Procedures • User Functions • User Sequences <p>These items can be imported as a Class for each Stored Procedure, User Function or User Sequence. Alternatively, all Stored Procedures can be imported as a set of operations contained under a Class. A similar Class can be created for all User Functions and User Sequences.</p> <p>Import as individual Classes</p> <p>The script is imported into separate Classes. The Properties dialog for each of these Classes supports a context sensitive editor for editing the script.</p> <p>Import as Operations</p> <p>The selected items are imported to a single Class as Operations (methods); you can view and edit them through the Operations Properties dialog of the parent Class.</p>	<p>Stored Procedures^[1967]</p>
7	<p>In the Synchronization panel, select the appropriate option to determine whether the existing Classes are to be updated, or the database objects imported as new objects.</p> <p>If you select the Synchronize existing classes option, also select the appropriate checkboxes to determine whether model comments, column default values and/or table constraints are to be retained or overwritten with the comments, values and constraints of the imported objects.</p>	
8	<p>Click on the Import button to start the import.</p>	
9	<p>Select the database objects to import.</p>	<p>Select Tables^[1976]</p> <p>Imported Class Elements^[1976]</p>

Step	Action	See also

Notes

- Enterprise Architect requires **32-bit ODBC** drivers to connect to a repository through ODBC; to set up the ODBC configuration on **64-bit** clients, run the 32-bit ODBC Data Source Administrator from `C:\Windows\SysWOW64\odbcad32.exe`
- The ODBC connection should use the ODBC driver available from the DBMS vendor, such as MySQL's ODBC driver for MySQL, and Oracle's ODBC driver for Oracle; drivers provided by third-party vendors are not supported, including the Microsoft ODBC driver for Oracle
- You can import a suitable ODBC driver for SQLite from <http://www.ch-werner.de/sqliteodbc/>
- If setting up a ODBC connection for reverse engineering, the default settings are sufficient
- If you are importing database schema from an MS Access Jet 4.0 database, please ensure that you have selected the **Use Jet 4.0** checkbox on the General page of the Options dialog, otherwise the Jet 3.5 routines are loaded; you must restart Enterprise Architect after selecting the checkbox
- Additional data types are available from the *Data Modeling Data Types* section of the Resources page on the Sparx Systems website
- If no diagram is open, the **Package Only** radio button defaults to selected and the options are disabled; if the open diagram is in the selected package, you can select either option

Learn more

- [Import Source Code](#)^[1790]
- [General Settings](#)^[544]
- [Data Modeling Data Types](#) (Online Resource)

Learning Center topics

- (Alt + F1) | Database Engineering | Import Schema**

11.2.1 Select a Data Source

Topics

Topic	Detail	See also
General Usage	To import DDL from existing data sources, you must have a suitable ODBC connection installed and configured. From the Import DB Schema from ODBC Source dialog you can select the ODBC data source using the standard windows ODBC set-up dialog; click on the data source name and then click on the OK button.	

Learn more

- [Connect to a Data Repository](#)^[209]

11.2.2 Select Tables

When you have opened the ODBC data source, Enterprise Architect acquires a list of database objects suitable for importing. This is presented in a list form for you to select from.

Highlight the schema to import from the top list. Alternatively, select individual objects from the lower list.

Topics

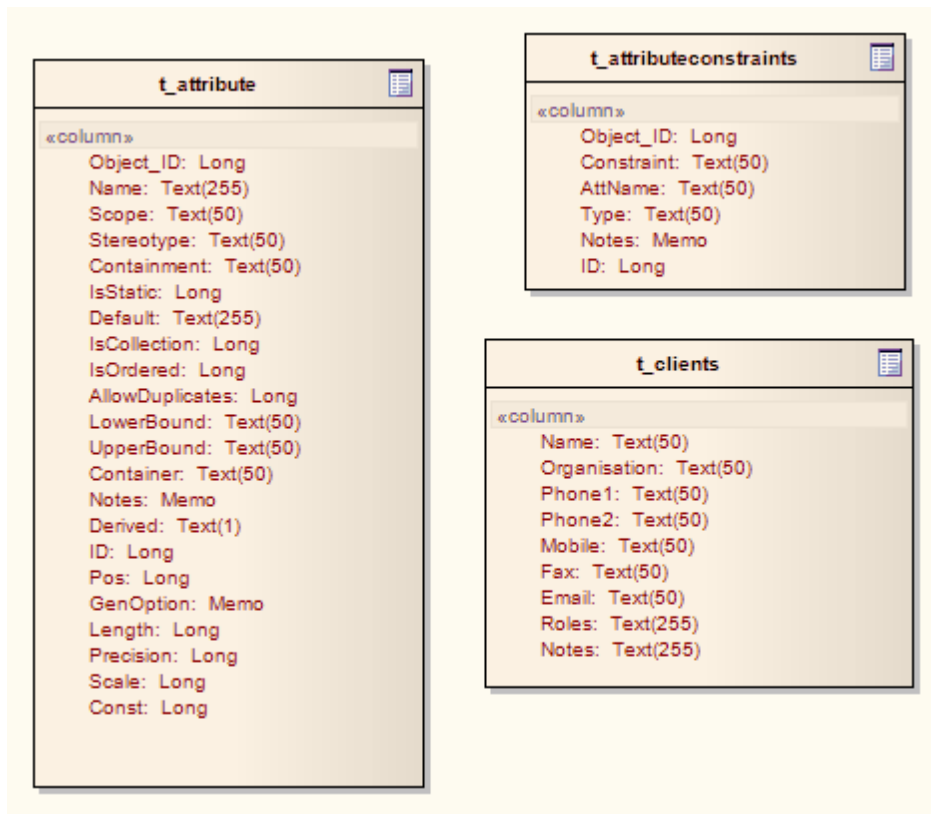
Topic	Detail	See also
Selection Shortcuts	<ul style="list-style-type: none">• To select all objects, click on the Select All button• To clear all objects, click on the Select None button• Hold down (Ctrl) while clicking to select multiple objects• Hold down (Shift) and click on the first and last objects in a range to select that range <p>When you have selected the objects, click on the OK button.</p>	

11.2.3 The Imported Class Elements

When you import DDL table definitions they are converted to stereotyped Classes according the *UML Data Modeling Profile*.

The image below shows some example tables imported into the model using an ODBC data connection.

Example



11.3 Generate DDL

Enterprise Architect enables you to generate DDL from your model for Tables, Stored Procedures and Views. SQL script can be generated for a single element (a Table) or for a whole package.

Having generated DDL, you can view it using the Source Code editors, by clicking on the table or package in the Project Browser or a diagram, and pressing **(F12)**, **(Ctrl+E)** or **(Alt+7)**.

The editor cannot parse DDL, and therefore does not show a structure tree for a DDL file in the left-hand panel of the Source Code window or view.

Learn more

- [Generate DDL for Tables](#)^[1978]
- [Generate DDL for Packages](#)^[1979]
- [Editing Source Code](#)^[1799]

Learning Center topics

- **(Alt + F1) | Database Engineering | Generate DDL**

11.3.1 Generate DDL For a Table

Access [Diagram Table element context menu | Generate DDL](#)

Reference

On the Generate DDL dialog, generate DDL for a table as indicated below:

Step	Action	See also
Path	Click on the (...) (Browse) button to set up the file location and file name of the script to create.	
Comment Level	To include comments in the DDL, select the appropriate level. For example, select Column to include comments on columns, or All to include comments on all structures.	
Option Checkboxes	Select the checkbox for each type of inclusion you require; for example, to include a 'drop table'. command in the script, select the Create Drop SQL checkbox. Deselect the checkboxes for inclusions you do not require.	
Generate	Click on this button to create the DDL.	
View	Click on this button to view the output.	Editing

Step	Action	See also
	The viewer defaults to the Enterprise Architect default code editor. However, you can define an alternative default DDL editor on the Options dialog (Tools Options Source Code Engineering Code Editors).	Source Code <small>[1799]</small> Options - Code Editors <small>[1895]</small>

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Generate Source Code and DDL** permission to generate DDL
- Some checkboxes display only if the appropriate database is defined for the table; for example, **IF EXISTS** displays only if the database for the table is PostgreSQL, and **SEQ_** and **_SEQ** radio buttons display only if the database for the table is Oracle
- For a PostgreSQL database, you must select the **Generate Sequences** checkbox to enable auto increment columns to be created
- If generating Oracle sequences, you must always select the **Generate Triggers** and **Generate Sequences** checkboxes - this ensures that a pre-insert trigger is generated to select the next sequence value to populate the column; also set the **AutoNum** property to **True** in the column properties
- If generating Oracle sequences, to generate the sequence name and trigger name with the syntax **SEQ_<sequence_name>** and **TRG_<trigger_name>**, select the **SEQ_** option; conversely, to generate them with the syntax **<sequence_name>_SEQ** and **SET_<trigger_name>**, select the **_SEQ** option

Learn more

- [Permission List](#)^[264]
- [Create Columns](#)^[1942]

11.3.2 Generate DDL for a Package

In this procedure, you can generate DDL for a package, and also compare the DDL with the database.

Access **Project Browser package context menu | Code Engineering | Generate DDL**
Select package Tools | Database Engineering | Generate Package DDL

How to

To generate DDL for a package

Step	Action	See also
1	On the Generate Package DDL dialog, select the checkbox against each inclusion required.	

Step	Action	See also
	Deselect the checkboxes for inclusions you do not require.	
2	To recursively generate DDL, select the Include All Child Packages checkbox.	
3	Select the Save Generated Order checkbox to save the order in which the objects are generated. This is useful where the order is changed to resolve object dependencies; the order is saved when you click on the Generate button.	
4	If you click the Refresh button with the Save Generated Order checkbox <i>unchecked</i> , the objects are restored to their original order.	
5	Click on the Generate button to proceed. Enterprise Architect prompts you for file names as the process executes.	
6	To view the output, click on the View button. The viewer defaults to the Enterprise Architect default code editor. However, you can define an alternative default DDL editor on the Options dialog (Tools Options Source Code Engineering Code Editors).	Editing Source Code <small>[1799]</small> Options - Code Editors <small>[1895]</small>

To compare the generated DDL with the database

Step	Action	See also
1	On the Generate Package DDL dialog, click on the Compare button. The Compare With Database dialog displays.	
2	Click on the (...) button and locate the required database on the Select Data Source dialog.	Set Up an ODBC Driver for a Connection to a Repository <small>[200]</small>
3	For an Oracle database, if required you can also specify the Owner in the Schema/Owner field.	
4	Click on the View button to perform the comparison. The Comparison Database dialog displays with the results of the comparison.	

Click on each table name to review information on that table.

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Generate Source Code and DDL** permission to generate DDL
- Some checkboxes display only if the appropriate database is defined for the table; for example, **IF EXISTS** displays only if the database for the table is PostgreSQL, and **SEQ_** and **_SEQ** radio buttons display only if the database for the table is Oracle
- For a PostgreSQL database, you must select the **Generate Sequences** checkbox to enable auto increment columns to be created
- If generating Oracle sequences, you must always select the **Generate Triggers** and **Generate Sequences** checkboxes - this ensures that a pre-insert trigger is generated to select the next sequence value to populate the column; also set the **AutoNum** property to **True** in the column properties
- If generating Oracle sequences, to generate the sequence name and trigger name with the syntax *SEQ_<sequence_name>* and *TRG_<trigger_name>*, select the **SEQ_** option; conversely, to generate them with the syntax *<sequence_name>_SEQ* and *SET_<trigger_name>*, select the **_SEQ** option

Learn more

- [Permission List](#)^[264]
- [Create Columns](#)^[1942]

11.4 Data Modeling Notations

Enterprise Architect supports three diagram notations for data modeling; UML 2.1, IDEF1X and Information Engineering.

Access [Diagram | Properties > Connectors | Connector Notations](#)

Topics

Topic	Detail	See also
UML 2.1	The standard UML 2.1 notation for connectors.	
Information Engineering	The Information Engineering (IE) connection style.	
IDEF1X	Integrated Definition Methods IDEFX1 connection style.	

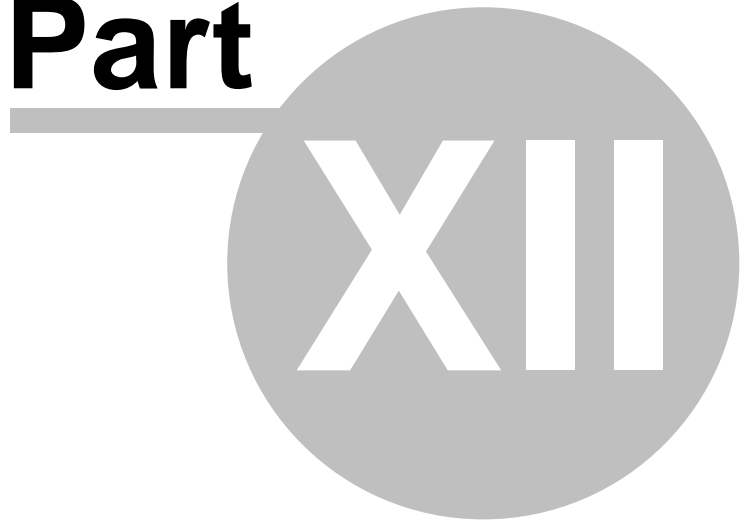
Notes

- The default notation for the Data Modeling diagram is Information Engineering

Learn more

- [Connectors Tab](#)⁷³²
- [Entity Relationship Diagrams \(ERDs\)](#)¹⁶²⁷

Part



XII

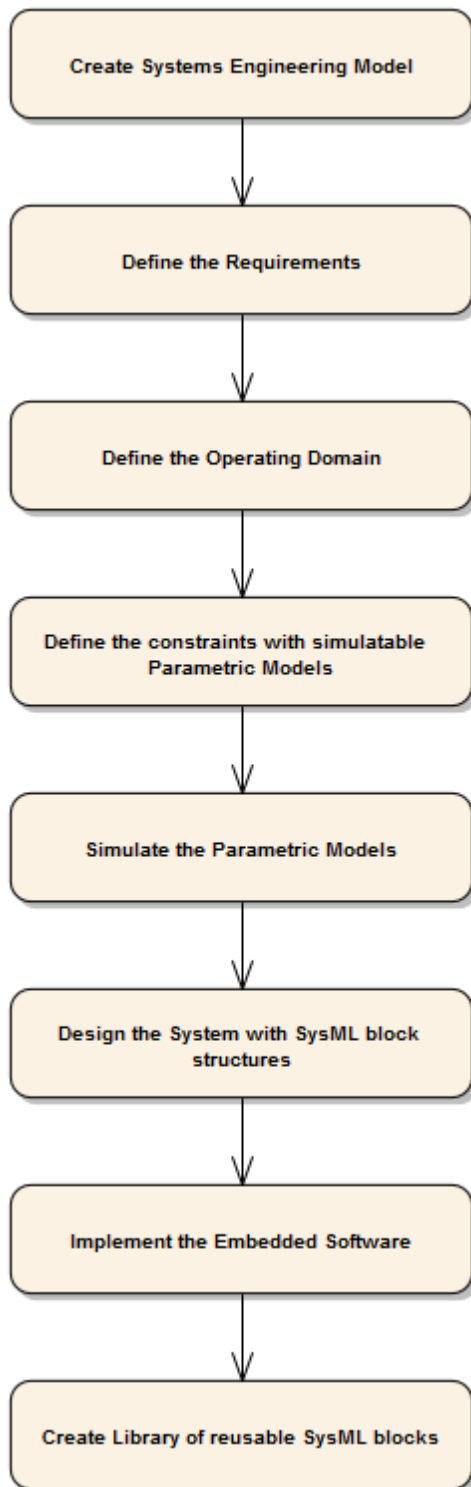
12 Systems Engineering

Topics

Topic	Detail	See also
Usage	<p>To model Systems using SysML in Enterprise Architect, you work through the following steps:</p> <ul style="list-style-type: none"> • Create a Systems Engineering model to develop your system • Create a Requirements model to define the systems requirements and expectations • Create an Operational Domain model, which describes the environment that the system operates within, and the entities it interacts with • Create Constraint models to describe the systems operating characteristics using parametric models • Simulate the parametric models to verify their correctness and obtain the desired characteristic • Design the system's composition using SysML Blocks and Parts • Implement the embedded software using UML Classes and behavioral models • Create a Library of reusable SysML blocks, representing subsystems that can be reused on other projects, and other common Type definitions 	<p>SysML^[1990]</p> <p>Create a Systems Engineering Model^[1985]</p> <p>A SysML Requirements Model^[2018]</p> <p>A SysML Operational Domain Model^[2019]</p> <p>SysML Parametric Models^[2013]</p> <p>Simulating a SysML Model^[2016]</p> <p>Compose System Design^[2021]</p> <p>Generate from Behaviour Models^[1778]</p> <p>Create Re-usable Subsystems^[2023]</p>

Example

These steps are represented graphically in the following flow:



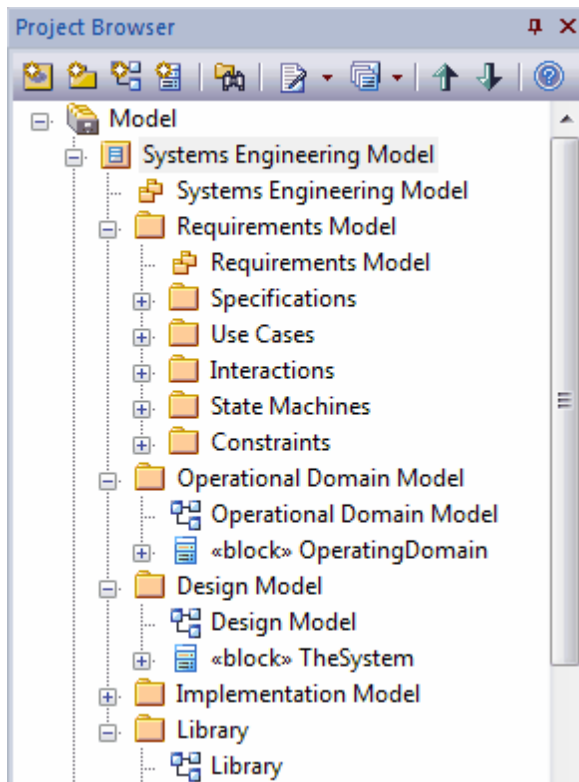
How to

To create a Systems Engineering Model from a template provided with Enterprise Architect

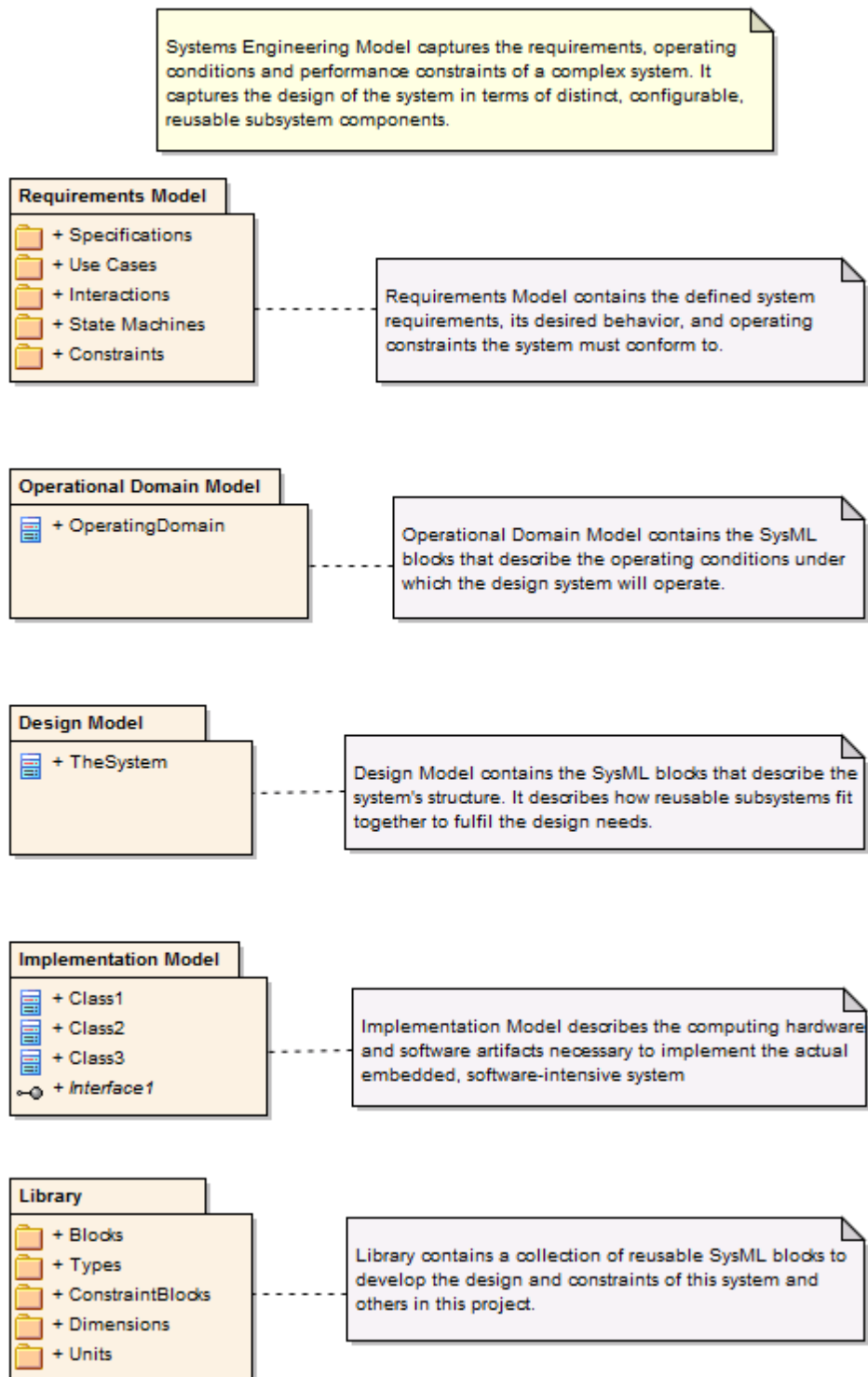
Step	Action	See also
1	<p>In the Project Browser, either:</p> <ul style="list-style-type: none">• Click on the New Model From Pattern icon in the toolbar• Right-click on a model root node and select the Add a New Model using Wizard context menu option, or• Right-click on a package and select the Add Add a New Model using Wizard context menu option <p>The Select Model(s) dialog displays.</p>	
2	<p>In the Select From field, click on the drop-down arrow and select Systems Engineering Model.</p> <p>Alternatively, if it is listed in the Technology panel, select the Systems Engineering Model item.</p>	
3	<p>In the Name panel, select the checkbox next to the Systems Engineering Model icon.</p>	
4	<p>Click on the OK button.</p>	

Example

The following model structure is created in the Project Browser:



The *Systems Engineering Model* diagram, shown below, encapsulates the key components of the Systems Engineering model.



Notes

- Systems Modeling Language (SysML) is provided with the Systems Engineering edition and Ultimate edition of Enterprise Architect

Learning Center topics

- (Alt+F1) | **Systems Engineering**

12.1 SysML

Topics

Topic	Detail	See also
Introduction	<p>The following text is derived from the official OMG SysML site of the Object Management Group</p> <p><i>The OMG Systems Modeling Language (OMG SysML™) is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities.</i></p> <p><i>The language provides graphical representations with a semantic foundation for modeling system requirements, behavior, structure, and parametrics, which is used to integrate with other engineering analysis models.</i></p> <p><i>SysML was developed in response to requirements developed jointly by the OMG and the International Council on Systems Engineering (INCOSE) by the diverse group of tool vendors, end users, academia, and government representatives.</i></p>	OMG SysML
SysML In Enterprise Architect	<p>Enterprise Architect enables you to develop SysML models quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer</p> <p>The SysML technology provides:</p> <ul style="list-style-type: none"> • Each of the nine SysML diagram types, accessed through the New Diagram dialog • A collection of SysML pages in the Toolbox that contain each of the supported SysML elements and relationships • SysML element and relationship entries in the Toolbox Shortcut Menu and Quick Linker • Simulation of SysML Parametric diagrams, which supports engineering analysis of critical system parameters including the evaluation of key metrics such as performance, reliability and other physical characteristics 	Add New Diagram ^[724] Toolbox Shortcut Menu ^[704] Quick Linker ^[791] Simulate a SysML Model ^[2016]
SysML Toolboxes	<p>You can access the SysML pages of the Toolbox through the More tools SysML 1.1 and SysML 1.2 menu options.</p> <p>You can also enable SysML as the active technology to access the Toolbox pages directly.</p>	Model Elements ^[2002] Block Definition ^[1996] Internal Block ^[2000] Parametrics ^[2005]

Topic	Detail	See also
	<p>The following sets of Toolbox pages are available:</p> <ul style="list-style-type: none"> • Model Elements contains the constructs required to build SysML models, package structures and views • Block Definition contains the constructs required to design SysML blocks, constraint blocks, data and value types • Internal Block contains the constructs required to design SysML block compositions within Internal Block Diagrams • Parametrics contains the constructs required to construct SysML Parametric Diagrams using constraint blocks • Activity contains the constructs required to construct SysML Activity models • Interaction contains the constructs required to construct SysML interactions and Sequence diagrams • State Machine contains the constructs required to build SysML State Machines • Use Case contains the constructs required to build SysML Use Case models • Requirements contains the constructs required to build SysML Requirements models <p>With the Model Elements pages there is a set of SysML Common elements and relationships; these are also provided with the other SysML Toolbox pages if the active technology is set to SysML 1.1 or SysML 1.2 on the Default Tools toolbar.</p>	<p>Activity^[1992]</p> <p>Interaction^[1999]</p> <p>State Machine^[2009]</p> <p>Use Case^[2011]</p> <p>Requirements^[2007]</p> <p>Default Tools Toolbar^[139]</p>
<p>Convert SysML 1.1 Models to SysML 1.2</p>	<p>Enterprise Architect enables you to migrate a SysML 1.1 model (or part of a model) to SysML 1.2, using the Automation Interface.</p>	<p>Migrate SysML 1.1 Model to SysML 1.2^[2024]</p>
<p>Disable SysML</p>	<p>If you prefer not to use SysML in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).</p>	<p>MDG Technologies^[1285]</p>

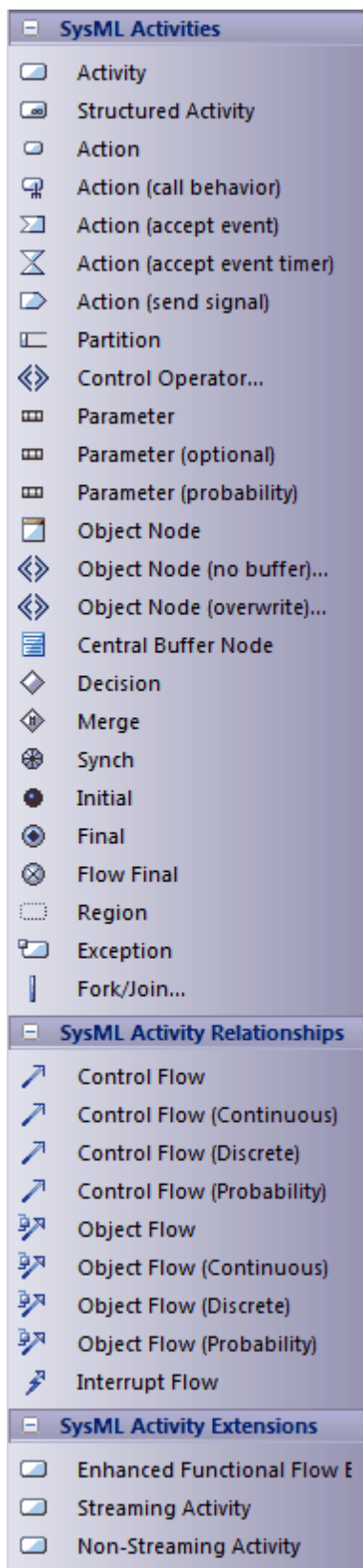
Notes

- Systems Modeling Language (SysML) is provided with the Systems Engineering and Ultimate editions of Enterprise Architect

12.1.1 SysML Activity

These pages provide the SysML elements to construct SysML Activity models.

Access [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML 1.2 Activity](#)



Page	Item	Action
SysML Activities	Activity	Define a SysML Block of executable behavior as a UML Activity.
	Structured Activity	Define a SysML Block of executable behavior as a UML Structured Activity.
	Action	Declare a unit of execution in an Activity as a UML Action.
	Action (call behavior)	Declare a unit of execution that calls another behavior.
	Action (accept event)	Declare a unit of execution that accepts an event raised by the system.
	Action (accept event timer)	Declare a unit of execution that accepts an event raised by a time epoch.
	Action (send signal)	Declare a unit of execution that sends a signal as an event.
	Partition	Create an Activity Partition to group execution elements together by the node responsible for their execution.
	Control Operator	Control the execution of an Activity.
	Parameter	Provide access to input and output objects within the Activity.
	Parameter (optional)	Define a parameter whose contents are optional in the Activity's execution.
	Parameter (probability)	Tag a parameter with probability of the likelihood of the parameter's use in the Activity.
	Object Node	Declare a variable in the Activity, typed by a ValueType, DataType or Block.
Object Node (no buffer)	Declare an ObjectNode in an Activity which discards unconsumed tokens.	
Object Node	Declare an ObjectNode in an Activity which overwrites tokens.	

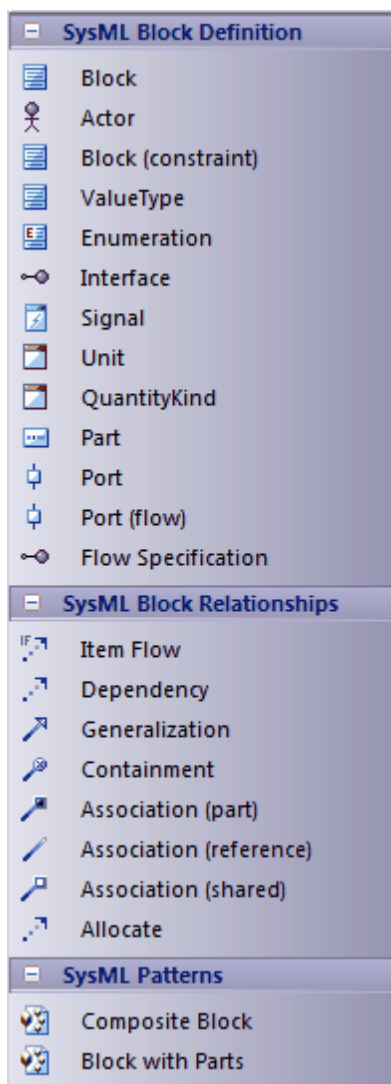
Page	Item	Action
	(overwrite)	
	Central Buffer Node	Declare an ObjectNode that stores tokens for consumption throughout the Activity.
	Decision	Create a branch of control in an Activity based on a decision.
	Merge	Merge two or more Activity control branches.
	Synch	Establish a rendezvous point for two or more Activity flows, in order to synchronize their execution in the Activity.
	Initial	Declare the start of Activity's execution.
	Final	Declare the end of an Activity's execution, and the termination of the Activity.
	Flow Final	Declare the end of an Activity's execution path without terminating the Activity.
	Region	Group a subset of an Activity into a common execution context.
	Exception	Declare a node of execution that happens outside the normal flow of execution of an Activity.
	Fork/Join	Simultaneously branch / join a set of Control or Object Flows.
SysML Activity Relationships	Control Flow	Establish a flow of logic between two Activity nodes.
	Control Flow (Continuous)	Declare a continuous control flow.
	Control Flow (Discrete)	Declare a discrete control flow.
	Control Flow (Probability)	Tag a control flow with a probability of the likelihood of the flow's traversal.
	Object Flow	Establish a flow of objects (data) between two Activity nodes.

Page	Item	Action
	Object Flow (Continuous)	Declare a continuous object flow.
	Object Flow (Discrete)	Declare a discrete object flow.
	Object Flow (Probability)	Tag an object flow with a probability of the likelihood of the flow's traversal.
	Interrupt Flow	Declare a control flow that interrupts flow within a Region.
SysML Activity Extensions	Enhanced Functional Flow Block Diagrams	Declare an Activity used to contain an Enhanced Functional Flow Block Diagram (EFFBD).
	Streaming Activity	Declare an Activity where the flow of tokens passes through its parameters continuously throughout the Activity's execution.
	Non-Streaming Activity	Declare an Activity where the flow of tokens passes through its parameters at the start of the Activity's execution.

12.1.2 SysML Block Definition

These pages provide the SysML elements to design SysML blocks, constraint blocks, data and value types.

[Access](#) [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML 1.2 Block Definition](#)



Page	Item	Action
SysML Block Definition	Block	Define a composite system entity in SysML.
	Actor	Represent a user that interacts with one or more SysML systems.
	Block (constraint)	Define a composite constraint as a system of parametric equations.
	ValueType	Define a SysML quantity, expressed as a measurable dimension with specific units.

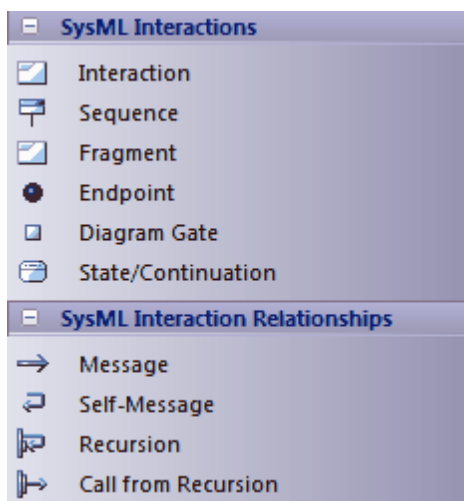
Page	Item	Action
	Enumeration	Define a data type as a set of symbols or values.
	Interface	Define an element that describes a specification of an interaction point with properties and methods.
	Signal	Define a SysML message, containing attributes, exchanged between system blocks in an interaction.
	Unit	Represent a standard unit of measure in SysML.
	QuantityKind	Identify a measurable quantity in SysML.
	Part	Describe the decomposition of a SysML Block subsystem in the context of its whole using instances of reusable SysML Blocks.
	Port	Describe a structural interaction point of a SysML Block which, in turn, connects between interacting parts of a block.
	Port (flow)	Describe what flows in and out of interacting SysML Blocks.
	Flow Specification	Define a set of flow properties that correspond to individual pieces of a common interaction point.
SysML Block Relationships	Item Flow	Specify the items that flow across a connector in an interaction point.
	Dependency	Establish a traceable relationship describing how one element is dependant upon another.
	Generalization	Describe an element as a specialized descendant of another element, containing additional properties and behavior.
	Containment	Graphically display ownership of one element within a parent element.
	Association (part)	Describe the characteristics of a connection between a SysML Block and its internal parts, such as the multiplicity and type.

Page	Item	Action
	Association (reference)	Describe the characteristics of a connection between separate SysML Blocks, such as the multiplicity and type.
	Association (shared)	Describe the characteristics of a common connection between SysML Blocks, such as the multiplicity and type.
	Allocate	Relate model elements together to formalize a refinement of behavior, structure, constraints or design expectations.
SysML Patterns	Composite Block	A pattern that creates a Composite Block made up of Blocks related by Aggregation relationships.
	Block with Parts	A pattern that creates a Composite Block made up of nested parts.

12.1.3 SysML Interaction

These pages provide the SysML elements to construct SysML interactions and Sequence diagrams.

[Access](#) [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML1.2 Interaction](#)



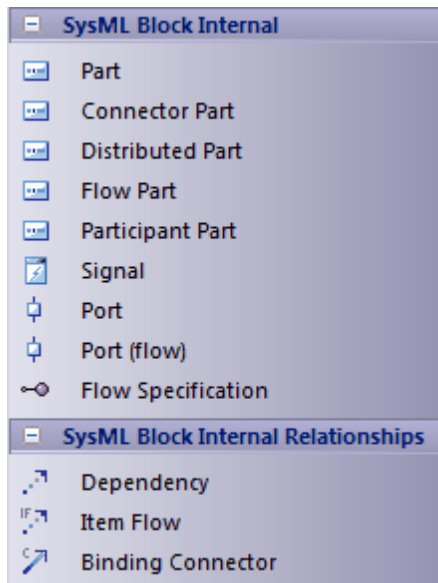
Page	Item	Action
SysML	Interaction	Define a SysML Block of executable behavior as a UML

Page	Item	Action
Interactions		Interaction.
	Sequence	Reference an instance of a SysML Block as a Lifeline in the Interaction.
	Fragment	Declare a portion of an interaction as a group with specific behavior. semantics.
	Endpoint	Create an exit point for the Interaction.
	Diagram Gate	Create an endpoint for the interaction, which bridges between nested interactions.
	State/Continuation	Constrain the Interaction with assertions of the state that the lifeline is expected to be in.
SysML Interactions Relationships	Message	Describe a message exchange between two lifelines in an Interaction.
	Self-Message	Describe a message exchange between a lifeline and itself in an Interaction.
	Recursion	Describe a recursive message exchange between a lifeline and itself in an Interaction.
	Call from Recursion	Describe a message exchange between two lifelines within a recursive exchange.

12.1.4 SysML Internal Block

These pages provide the SysML elements to design SysML block compositions within Internal Block Diagrams.

[Access](#) [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML 1.2 Internal Block](#)



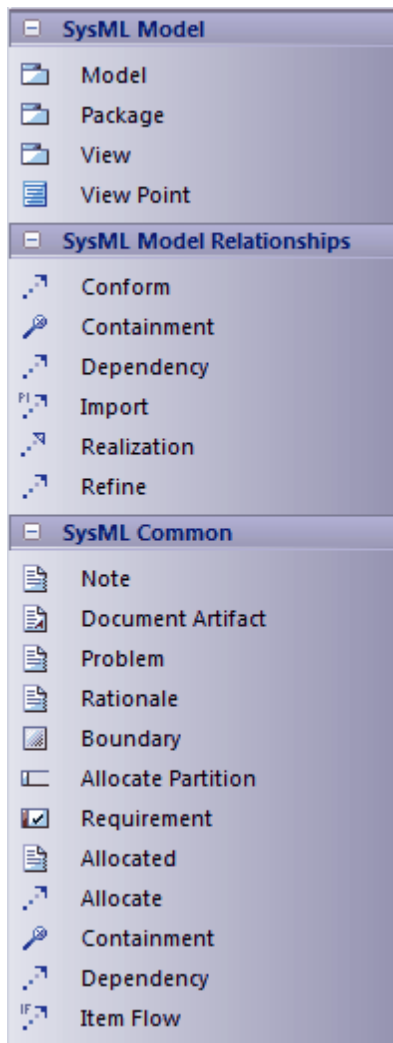
Page	Item	Action
SysML Block Internal	Part	Describe the decomposition of a SysML Block subsystem in the context of its whole, using instances of reusable SysML Blocks.
	Connector Part	Create a SysML connector part.
	Distributed Part	Create a SysML distributed part.
	Flow Part	Create a SysML flow part.
	Participant Part	Create a SysML participant part.
	Signal	Define a SysML message, containing attributes, exchanged between system blocks in an interaction.
	Port	Describe a structural interaction point of a SysML Block which, in turn, connects between interacting parts of a block.
	Port (flow)	Describe what flows in and out of interacting SysML Blocks.
	Flow Specification	Define a set of flow properties that correspond to individual pieces of a common interaction point.

Page	Item	Action
SysML Block Internal Relationships	Dependency	Establish a traceable relationship describing how one element is dependant upon another.
	Item Flow	Specify the items that flow across a connector in an interaction point.
	Binding Connector	Establish a connection between two parts in a system decomposition.

12.1.5 SysML Model Elements

These pages provide the SysML elements to build SysML models, package structures and views.

[Access](#) [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML 1.2 Model Elements](#)



Page	Item	Action
SysML Model	Model	Create a Package containing a SysML Model.
	Package	Group model constructs together in a single unit of containment.
	View	Create a stereotyped Package that defines a SysML View of a system, from the perspective of a SysML View Point.
	View Point	Create a stereotyped Class that defines a SysML View Point, which specifies the rules and conventions for the construction and use of Views.

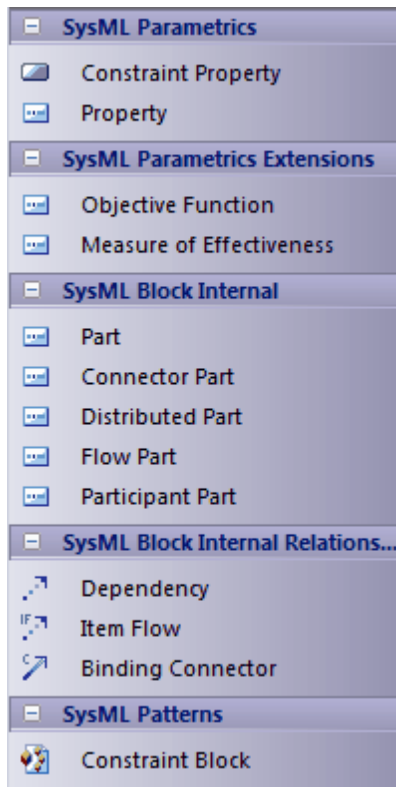
Page	Item	Action
SysML Model Relationships	Conform	Establish a conformance dependency of a View to the defining View Point.
	Containment	Graphically display ownership of one element within a parent one.
	Dependency	Establish a traceable relationship describing how one element is dependant upon another.
	Import	Represent a reuse of elements from one model package in another.
	Realization	Identify a design fulfillment of a specification between elements.
	Refine	Represent a refinement of one element by another.
SysML Common	Note	Create a textual annotation that can be attached to a set of elements of any other type. The attachment is created separately, using a Notelink connector.
	Document Artifact	Attach a linked document to the diagram by associating this element with an RTF document.
	Problem	Document the failure of model elements to satisfy a requirement.
	Rationale	Document the justification for decisions.
	Boundary	Define a conceptual boundary, to visually group logically related elements.
	Allocate Partition	Contains elements deemed to be allocated to the classifier of the partition.
	Requirement	Specify the capabilities of the system, or the conditions that it should satisfy.

Page	Item	Action
	Allocated	The source element being allocated to the target element in an Allocate relationship.
	Allocate	Relate model elements to formalize a refinement of behavior, structure, constraints or design expectations. The Allocate relationship points from the element being allocated to the element that is the target of the allocation.
	Containment	Graphically display ownership of one element within a parent one.
	Dependency	Establish a traceable relationship describing how one element is dependant upon another.
	Item Flow	Specify the items that flow across a connector in an interaction point.

12.1.6 SysML Parametrics

These pages provide the SysML elements to construct SysML Parametric Diagrams using constraint blocks.

[Access](#) [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML 1.2 Parametrics](#)



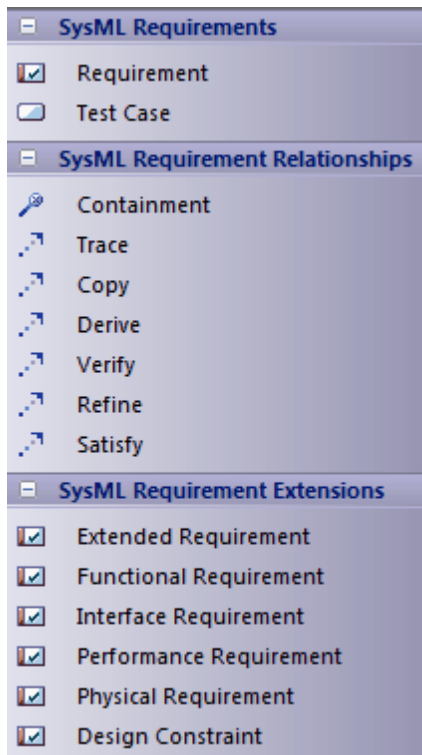
Page	Item	Action
SysML Parametrics	Constraint Property	Instantiate a Constraint Block for use in a Parametric diagram.
	Property	Define a SysML property typed by a DataType, ValueType or Block.
SysML Parametrics Extensions	Objective Function	Define a SysML Constraint Block for use as an objective function to evaluate Measures of Effectiveness (MOEs).
	Measure of Effectiveness	Define a SysML property for use as a Measure of Effectiveness (MOE).
SysML Block Internal	Part	Describe the decomposition of a SysML Block subsystem in the context of its whole, using instances of reusable SysML Blocks.
	Connector Part	Create a SysML connector Part.

Page	Item	Action
	Distributed Part	Create a SysML distributed Part.
	Flow Part	Create a SysML flow Part.
	Participant Part	Create a SysML participant Part.
SysML Block Internal Relationships	Dependency	Establish a traceable relationship describing how one element is dependant upon another.
	Item Flow	Specify the items that flow across a connector in an interaction point.
	Binding Connector	Establish a connection between two Parts in a system decomposition.
SysML Patterns	Constraint Block	A pattern that creates a typical Constraint property that owns two Parts.

12.1.7 SysML Requirements

These pages provide the SysML elements to build SysML Requirements models.

[Access](#) [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML 1.1 Requirements](#)



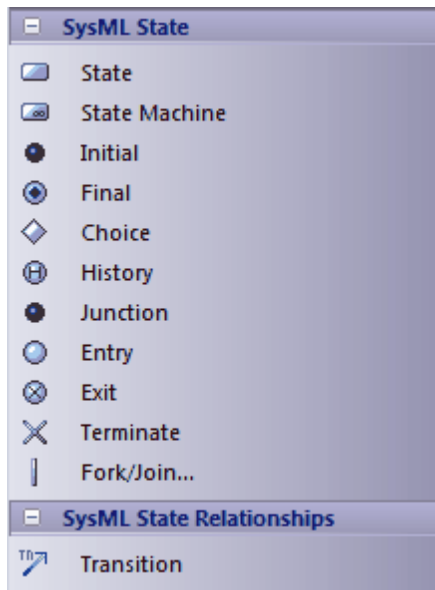
Page	Item	Action
SysML Requirements	Requirement	Specify the capabilities of the system, or the conditions that it should satisfy.
	Test Case	Describe the verification of a Requirement through methods of inspection, analysis, demonstration or testing.
SysML Requirement Relationships	Containment	Graphically display ownership of one element within a parent element.
	Trace	Declare a trace relationship between a SysML Requirement and another SysML element.
	Copy	Declare a copy of one SysML Requirement by another.
	Derive	Derive a SysML Requirement from another.
	Verify	Declare a verification of a SysML Requirement by another SysML element.

Page	Item	Action
	Refine	Declare a refinement of a SysML Requirement by another SysML element.
	Satisfy	Declare that the SysML Requirement is satisfied by another SysML element.
SysML Requirement Extensions	Extended Requirement	Extend a SysML Requirement with additional Tag properties.
	Functional Requirement	Declare a SysML Requirement that describes the operation, or behavior, that the system must perform.
	Interface Requirement	Declare a SysML Requirement that describes how the system connects, or interfaces with, other systems.
	Performance Requirement	Declare a SysML Requirement that describes how the system performs against defined capabilities or conditions.
	Physical Requirement	Declare a SysML Requirement that describes the physical characteristics, or physical constraints, of the system.
	Design Requirement	Declare a SysML Requirement that specifies a constraint on the implementation of the system.

12.1.8 SysML State Machine

These pages provide the SysML elements to build SysML State Machines.

[Access](#) [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML 1.2 State Machine](#)



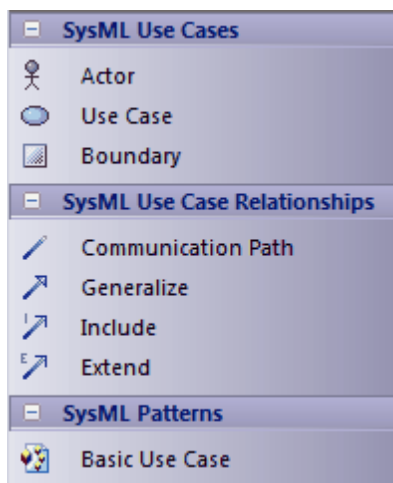
Page	Item	Action
SysML State	State	Declare a significant condition in the life of a SysML Block within its State Machine.
	State Machine	Describe the life-cycle behavior of a SysML Block in terms of its states and transitions.
	Initial	Declare the starting state of the State Machine.
	Final	Declare the ending state of the State Machine, and its completion.
	Choice	Declare a Junction with a mandatory 'else' transition.
	History	Represent the last active State of the State Machine prior to its interruption.
	Junction	Declare a decision point at which a Transition branches out into multiple guarded, alternative paths.
	Entry	Declare an Entry point between State Machines, Substate Machines and Regions.

Page	Item	Action
	Exit	Declare an Exit point between State Machines, Substate Machines and Regions.
	Terminate	Declare a termination State in which the State Machine no longer operates.
	Fork/Join	Simultaneously branch / join a set of Transitions.
SysML State Relationships	Transition	Establish a life-cycle path between one State and another, based on its operational conditions.

12.1.9 SysML Use Case

These pages provide the SysML elements to build SysML Use Case models.

[Access](#) [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML 1.2 Use Cases](#)



Page	Item	Action
SysML Use Cases	Actor	Represent a user that interacts with one or more SysML systems.
	Use Case	Describe the expected functionality of a system as a UML Use Case.
	Boundary	Graphically bound elements in a diagram with a border.

Page	Item	Action
SysML Use Case Relationships	Communication Path	Declare which Actors perform in the Use Case.
	Generalize	Describe an element as a specialized descendant of another element, containing additional properties and behavior.
	Include	Describe one Use Case as a subset of another.
	Extend	Describe one Use Case as an extension of another.
SysML Patterns	Basic Use Case	A pattern that creates a typical simple Use Case diagram of Actor, Use Case and System Boundary.

12.2 SysML Parametric Models

SysML Parametric models support the engineering analysis of critical system parameters, including the evaluation of key metrics such as performance, reliability and other physical characteristics.

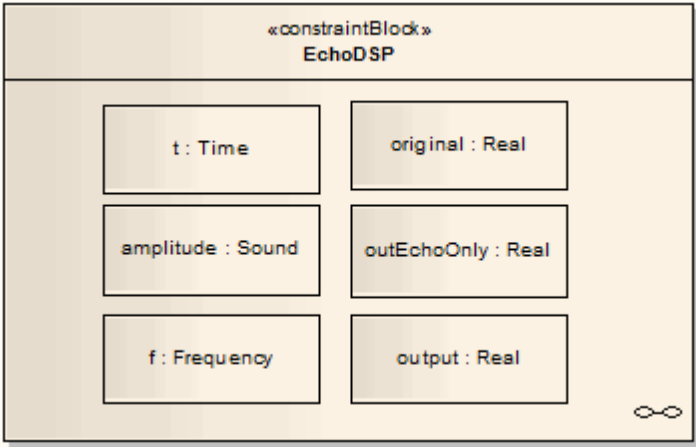
Topics

Topic	Detail	See also
SysML Parametric Models - Concept	<p>SysML Parametric Models unite requirements models with system design models by capturing executable constraints based on complex mathematical relationships.</p> <p>The following text is derived from the SysML entry in the online Wikipedia.</p> <p><i>The advantages of SysML over UML for systems engineering become obvious if you consider a concrete example, such as modeling an automotive system. With SysML you can use Parametric diagrams to precisely define performance and mechanical constraints such as maximum acceleration, curb weight, air conditioning capacity, and interior cabin noise management.</i></p> <p>For further information on the concepts of SysML Parametric models, refer to the official OMG SysML website and its linked sources.</p> <p>Enterprise Architect enables you to develop SysML Parametric models quickly and simply; these models can also be simulated.</p>	<p>SysML (Online Resource)</p> <p>OMG SysML (Online Resource)</p>

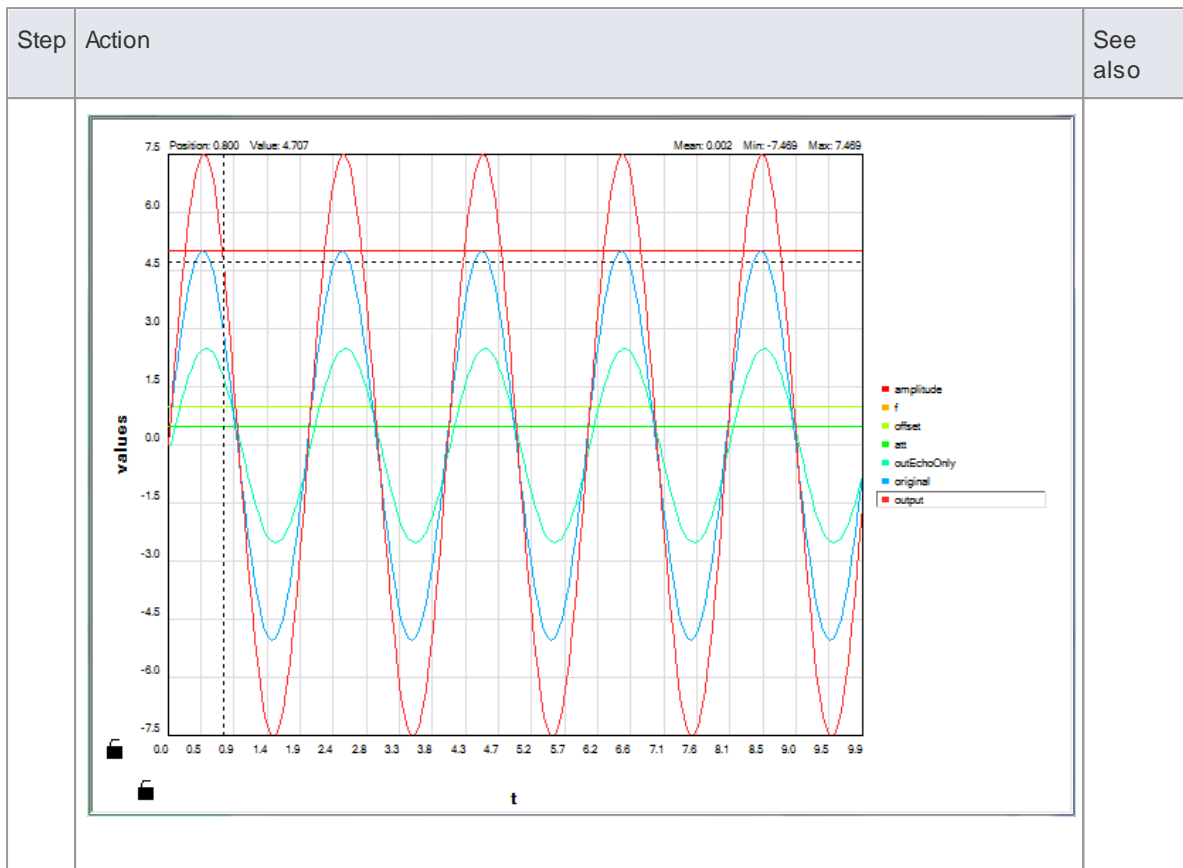
How to

To create a Parametric model

Step	Action	See also
1	<p>Create a collection of <i>SysML Constraint Blocks</i> that formally describe the function of a constraint in a simulation model.</p> <p>Each Constraint Block contains properties that describe its input and output parameters, as well as <i>Element Script</i> that describes the constraint's executable component.</p>	

Step	Action	See also
	 <p>The diagram shows a SysML Parametric Diagram for a constraint block named «constraintBlock» EchoDSP. The diagram is enclosed in a rectangular frame with a light beige background. At the top, the text «constraintBlock» and EchoDSP is centered. Below this, there are six rectangular boxes arranged in a 3x2 grid. The top row contains 't : Time' and 'original : Real'. The middle row contains 'amplitude : Sound' and 'outEchoOnly : Real'. The bottom row contains 'f : Frequency' and 'output : Real'. A small infinity symbol is located in the bottom right corner of the diagram frame.</p>	
2	<p>Right-click on each of the <i>constraintBlocks</i> and select the SysML Add Element Script context menu option to add script to the constraint block.</p> <p>This is where you express the relationship / behavior of the constraint block as an executable script.</p>	
3	<p>Create a SysML Constraint Block to contain the Parametric model to simulate; the Parametric model contains properties and occurrences of constraint blocks as Constraint Property elements, connected in a Parametric Diagram.</p>	

Step	Action	See also
	<p>The diagram illustrates a signal processing pipeline. On the left, a green dashed box labeled 'Inputs' contains five input boxes: 'amplitude: Sound', 'f: Frequency', 't: Time', 'att: Real', and 'offset: Real'. The 'SineWave' block takes 'f: Frequency' and 't: Time' as inputs and outputs a 'Real' value. The 'Mult' block takes the output of 'SineWave' and 'att: Real' as inputs. The 'Delay' block takes the output of 'Mult' and 'offset: Real' as inputs. The output of 'Delay' goes to a 'Buffer' block, which also receives the output of 'SineWave'. The output of this 'Buffer' block is 'original: Real'. The output of 'Delay' also goes to another 'Buffer' block, which outputs 'outEchoOnly: Real'. Finally, the outputs of both 'Buffer' blocks are fed into an 'Add2' block, which outputs 'output: Real'.</p>	
4	<p>Right-click within a Parametric Diagram and select the SysML Simulate Diagram... context menu option.</p>	
5	<p>Depending on your configuration selections, the simulation's results are either written to a comma-separated CSV file or graphed in a 2-dimensional plot.</p>	



Notes

- Systems Modeling Language (SysML) is provided with the Systems Engineering and Ultimate editions of Enterprise Architect

Learn more

- [SysML](#)^[1990]
- [Simulating a SysML Model](#)^[2016]

12.2.1 Simulate a SysML Model

You simulate a SysML model from a Parametric diagram, using the Simulation Configuration dialog.

Access [Parametric Diagram Context Menu | SysML | Simulate Diagram..](#)

How to

To simulate a SysML model

Step	Action	See also
1	The Parameters panel lists all of the parameters that can be assigned input.	

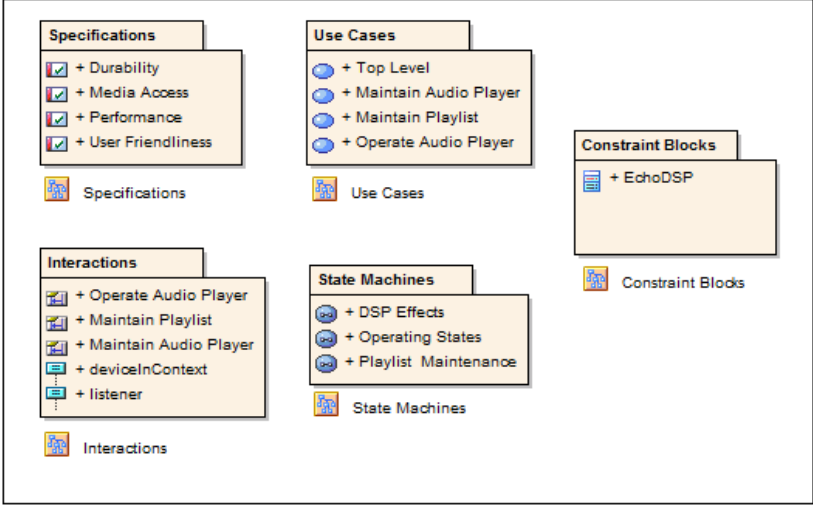
Step	Action	See also
	<p>Select each of the required parameters and click on the right Arrow button to assign them as input.</p> <p>Parameters designated as input parameters are listed in the Inputs panel on the right.</p> <p>There must be at least one input parameter assigned for the simulation to execute.</p>	
2	<p>Assign a set of values for each of the designated input parameters.</p> <p>For each input parameter, in the Input Values panel select one of the two possible value kinds:</p> <ul style="list-style-type: none"> • Discrete - To enter a constant or a comma-separated range of discrete values • Range - To enter a range of values beginning at the From value and ending at the To value; the input values are incremented by the Step value 	
3	<p>Specify the classes of output value:</p> <ul style="list-style-type: none"> • Parameters - To output the parameters' data, select the checkbox • Variables - To output the data generated within each internal variable, select the checkbox; internal variables are automatically generated by the simulator 	
4	<p>Specify how the simulation results are to be reported.</p> <p>The Output Format panel enables you to choose how the simulation outputs the simulation data:</p> <ul style="list-style-type: none"> • Plot To Graph: To plot the results on a 2-dimensional graph, select the checkbox; if you select this option, you must specify an input parameter for the plot's X Axis • Title - To enter a title for the graph, type in the title text • Output to File - To output the results to a CSV text file, select the checkbox and type or browse (...) for the file name 	
5	<p>Click on the OK button to execute the simulation.</p>	

Notes

- Systems Modeling Language (SysML) is provided with the Systems Engineering and Ultimate editions of Enterprise Architect

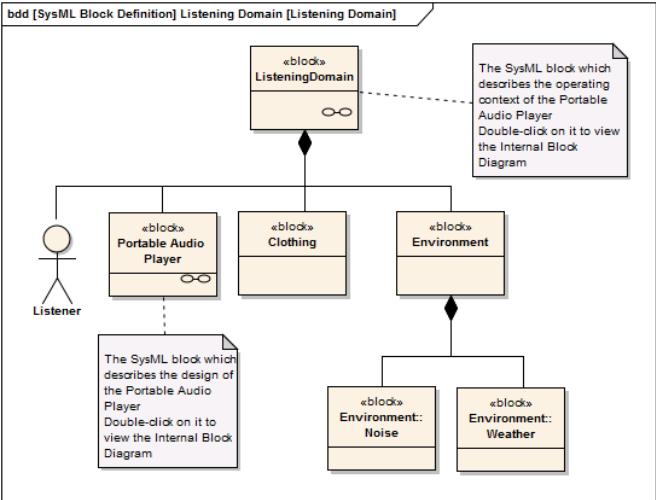
12.3 A SysML Requirements Model

Topics

Topic	Detail	See also
<p>Example</p>	<p>The SysML <i>Requirements Model</i> provides the system requirements, the expected abstract behavior and the operating constraints that the designed system must conform to.</p> <p>The following diagram shows an example requirements model for a <i>Portable Audio Player</i>.</p>  <p>In the example Requirements Model, each of the child packages contains child models that capture the following aspects of the system's requirements:</p> <ul style="list-style-type: none"> • The <i>Specifications</i> package contains SysML Requirements describing the overall expectations of the designed system • The <i>Use Cases</i> package contains SysML Use Cases that describe the general interaction between the system and its users • The <i>Interactions</i> package contains SysML Interactions that describe a detailed sequence of interactions between the system and its users • The <i>State Machines</i> package contains SysML State Machines that describe each of the operational states the designed system has • The <i>Constraint Blocks</i> package contains SysML ConstraintBlocks that describe the expected performance and operating boundaries of the system 	

12.4 A SysML Operational Domain Model

Topics

Topic	Detail	See also
<p>Abstract</p>	<p>The SysML <i>Operational Domain Model</i> defines the system's operating environment, which describes the operating conditions that the system is intended to operate within.</p> <p>The following diagram shows an example Operational Domain model for a <i>Portable Audio Player</i>; the SysML Block Definition Diagram describes the Operational Domain (in this example - the <i>ListeningDomain</i>) as a system composition.</p>  <p>The diagram is a SysML Block Definition Diagram (bdd) titled "Listening Domain [Listening Domain]". It shows a composition of several blocks: <ul style="list-style-type: none"> ListeningDomain: An «block» containing a composition of other blocks. Portable Audio Player: An «block» that is part of the ListeningDomain composition. It has a dashed line to a note: "The SysML block which describes the design of the Portable Audio Player. Double-click on it to view the Internal Block Diagram". Clothing: An «block» that is part of the ListeningDomain composition. Environment: An «block» that is part of the ListeningDomain composition. It has a composition of two sub-blocks: <ul style="list-style-type: none"> Environment::Noise: An «block». Environment::Weather: An «block». Listener: A stick figure actor connected to the Portable Audio Player block. </p> <p>In this example, the <i>ListeningDomain</i> is defined as a system containing other subsystems within it; the domain contains subsystems that define the Listener (i.e. User), the Portable Audio Player, Clothing (which the user wears), and the External Environment.</p> <p>Details of the <i>ListeningDomain</i> system are further detailed in the ListeningDomain's Internal Block Diagram:</p>	

Topic	Detail	See also
	<div data-bbox="507 318 1161 981" data-label="Diagram"> <p>ibd [SysML Internal Block] ListeningDomain [ListeningDomain]</p> <p>:Listener</p> <p>portableAudioPlayer</p> <p>listenerClothing</p> <p>externalNoise</p> <p>weather</p> <p>listeningConditions</p> </div> <p data-bbox="497 1041 1168 1220">In this example, the <i>ListeningDomain's</i> system's detailed composition shows how the Portable Audio Player and other sub-systems fit together to form the Listening Domain; it also describes the binding relationships between the parts, which describe how the parts are functionally bound to one another.</p>	

12.5 Compose System Design

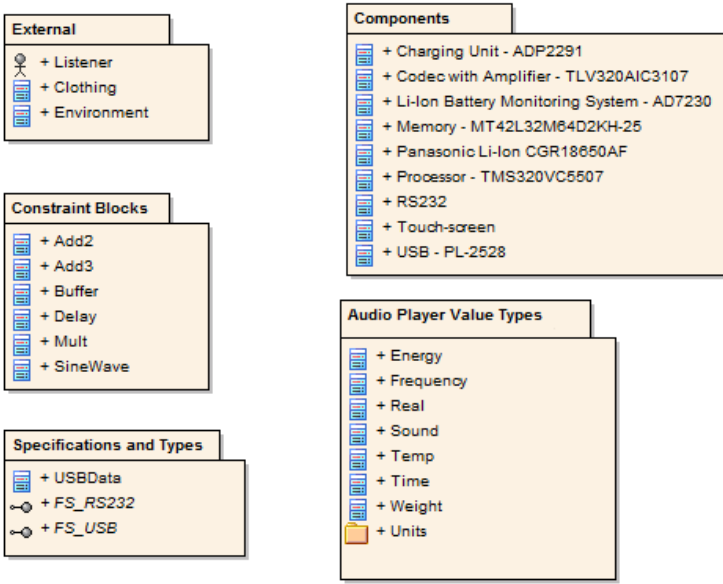
Topics

Topic	Detail	See also
<p>SysML Design Model</p>	<p>The <i>SysML Design Model</i> contains the blocks that define the system's composition; it describes the manner in which reusable subsystems fit together to fulfill the design requirements.</p> <p>The following diagram shows an example Design Model for a <i>Portable Audio Player</i>; the SysML <i>Block Definition Diagram</i> describes the <i>Portable Audio Player</i> as a composition of various reusable off-the-shelf subsystems and in-house designed ones.</p> <pre> graph TD subgraph "SysML Block Definition Diagram" direction BT PAA[Portable Audio Player] PAA --- par PPS[Portable Audio Player:PowerSubsystem] PAA --- proc PAS[Portable Audio Player:ProcessingSubsystem] PAA --- ui PAUI[Portable Audio Player:User Interface] PAA --- tr PATS[Portable Audio Player:TransportSubsystem] PPS --- bat PPS_BAT[Panasonic Li-Ion CGR18650AF] PPS --- chg PPS_CHG[Charging Unit - ADP2291] PPS --- pmon PPS_PMON[Li-Ion Battery Monitoring System - AD7230] PAS --- cpu PAS_CPU[Processor - TMS320VC5507] PAS --- oodec PAS_OODEC[Codec with Amplifier - TLV320AIC3107] PAS --- mem PAS_MEM[Memory - MT42L32M64D2KH-25] PAUI --- tsor PAUI_TSOR[Touch-screen] PAUI --- btin PAUI_BTIN[Portable Audio Player:User Interface:Buttons] PATS --- rst PATS_RST[RS232] PATS --- utr PATS_UTR[USB - PL-2528] end </pre> <p>In the example above, the <i>Portable Audio Player</i> is defined as a SysML system containing subsystems that perform specific tasks; the design contains subsystems for supplying power, performing playback and audio processing, interfacing with other devices and the user interface.</p> <p>Details of the <i>Portable Audio Player</i>'s composition are further described in detail within the <i>Portable Audio Player's Internal Block Diagram</i>:</p>	

Topic	Detail	See also
	<p>In this example, the <i>Portable Audio Player's</i> composition is described, detailing how each of the sub-systems is structured.</p> <p>It also describes the binding relationships between the parts, which describe how they are functionally bound to one another; for example, the CPU, Memory and Codec are interfaced together in the Processing Subsystem.</p>	

12.6 Create Reusable Subsystems

Topics

Topic	Detail	See also
SysML Design Model	<p>Model Based Systems Engineering provides the flexibility and expressiveness to define complex systems quickly and effectively, by reusing common entities across design projects.</p> <p>A <i>Library</i> is a package containing many reusable subsystems, parametric constraints, common data types and common value types, dimensions and units.</p> <p>The following diagram shows an example library model:</p>  <p>In the example Library, each of the child packages contains child models that capture the following reusable entities:</p> <ul style="list-style-type: none"> • <i>Blocks</i> defining systems such as those listed in the <i>Components</i> package, or those defined in the <i>External</i> package • <i>ConstraintBlocks</i> defining parametric constraints for use in parametric models. • <i>Value Types</i> describing quantities, expressed as measurable dimensions in specific units • <i>Data Types</i> and <i>Flow Specifications</i> describing data structures and <i>Flows</i> 	

12.7 Migrate SysML 1.1 Model to SysML 1.2

Enterprise Architect enables you to migrate a SysML 1.1 model (or part of a model) to SysML 1.2, using the Automation Interface function *Migrate()*. This function updates the Tagged Values and, if required, stereotypes to SysML1.2 for all elements, attributes, connectors and diagrams under the selected package or element.

Example

The following VB script calls the *Migrate()* function to migrate the package or element to SysML 1.2:

```
Sub MigrateElement (sGUID, lngPackageID)
    Dim proj as EA.Project
    set proj = Repository.GetProjectInterface
    proj.Migrate sGUID, "SysML1.1", "SysML1.2"

    'refresh the model
    If lngPackageID <> 0 Then
        Repository.RefreshModelView (lngPackageID)
    End If
End Sub

Sub MigrateSelectedItem
    Dim selType
    Dim selElement as EA.Element
    Dim selPackage as EA.Package
    selType = GetTreeSelectedItemType
    If selType = 4 Then 'means Element
        set selElement = GetTreeSelectedObject
        MigrateElement selElement.ElementGUID, selElement.PackageID
        MsgBox "Element Migration Completed", 0, "SysML Migration"
    ElseIf selType = 5 Then 'means Package
        set selPackage = GetTreeSelectedObject
        MigrateElement selPackage.PackageGUID, selPackage.PackageID
        MsgBox "Package Migration Completed", 0, "SysML Migration"
    Else
        MsgBox "Select a Package or Element in the Project Browser to
        initiate migration", 0, "SysML Migration"
    End If
End Sub

Sub Main
    MigrateSelectedItem
End Sub

Main
```

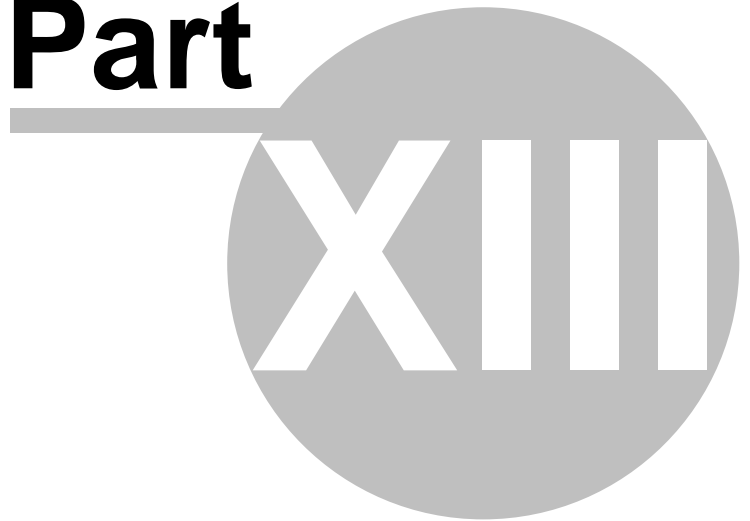
Notes

- The stereotype *dimension* is changed to *quantitykind* when migrating from SysML1.1 to SysML1.2
- The stereotype *dataType* is removed from SysML1.2
- The Tagged Value *dimension* is migrated to *quantitykind*; this applies to stereotypes *unit* and *valueType*
- The Tagged Value *isConjugated* of stereotype *flowport* is migrated to custom properties

Learn more

- [Migrate\(\)](#)^[2553]

Part



13 SOA and XML



This section explains how to import and generate:

- XSD
- WSDL

It also describes how to generate and export *Meta-Object* Facility (**MOF**) models, and the **SoaML** and **SOMF** technology integration.

Learn more

- [XML Schema](#)^[2027]
- [Web Services - WSDL](#)^[2067]
- [SoaML](#)^[2088]
- [SOMF](#)^[2092]
- [Generate MOF](#)^[2094]

Learning Center topics

- **(Alt+F1) | SOA and XML Engineering**

13.1 XML Schema - XSD

XML Schema Definition (XSD), also known as **XML Schema**, is a **W3C** XML technology that is used to specify the rules to which an XML document must adhere.

Enterprise Architect enables rapid modeling, forward engineering and reverse engineering of XML Schema. XSD support is critical for the development of a complete *Service Oriented Architecture* (SOA), and the coupling of UML 2.3 and XML provides the natural mechanism for specifying, constructing and deploying XML-based SOA artifacts within an organization.

Learn more

- [W3C XML Schema](#) (Online Resource)
- [Model XSD using Toolbox](#)^[2027]
- [Abstract XSD Models](#)^[2048]
- [Generate XSD](#)^[2049]
- [Import XSD](#)^[2053]
- [UML Profile for XSD](#)^[2055]

Learning Center topics

- (Alt+F1) | [SOA and XML Engineering | XSD](#)

13.1.1 Model XSD

XML schemas are modeled using UML **Class** diagrams. The XML Schema page of the Toolbox provides built-in support for the **UML profile for XSD**. This enables an abstract UML Class model to be automatically generated as a **W3C XML Schema** (XSD) file.

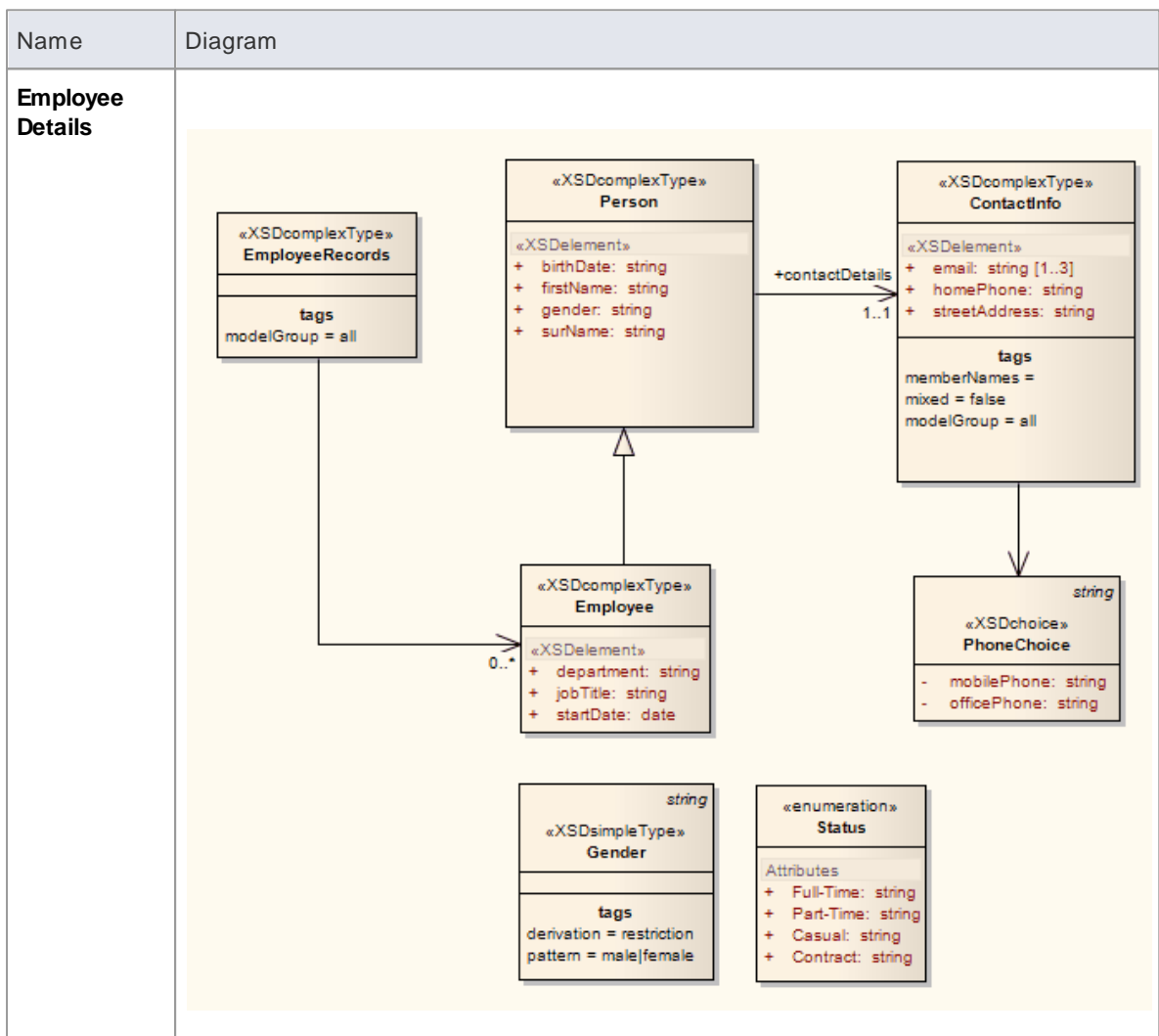
Access [View | Diagram Toolbox: More tools | XML Schema](#)

How to

Step	Action	See also
1	Create a package in the Project Browser.	
2	Click on the New Diagram icon in the Project Browser.	Project Browser Toolbar ^[583]
3	Select UML Structural in the Select From field and Class in the Diagram Types field.	
4	Click on the More tools option in the Toolbox and select XML Schema from the list of Toolboxes.	Diagram Toolbox ^[698]

<p>5</p>	<p>Click on the Schema icon from the Toolbox and drag it into the created Class diagram.</p> <p>When the New Diagram dialog is displayed, Select UML Structural in the Select From field and Class in the Diagram Types field to create a Class diagram.</p>	<p>Schema Icon ^[2029]</p>
<p>6</p>	<p>Enterprise Architect creates a new <i>XSDschema</i> stereotyped package with a Class diagram under it.</p> <p>Open this Class diagram and use the constructs from the XML Schema Toolbox to model the XML Schema.</p> <p><i>Alternatively:</i></p> <p>XML Schemas can also be modeled using the UML Profile for XSD.</p>	<p>XML Schema Group ^[720]</p> <p>UML Profile for XSD ^[2055]</p>

Example



Name	Diagram

Notes

- The UML attributes of the Classes map directly to XML elements or attributes
- Classes have no methods since there is no meaningful correspondence between Class methods and XSD constructs

Modeling Restrictions:

The following XML Schema constructs cannot be modeled in Enterprise Architect:

- *appinfo*
- *field*
- *key*
- *keyref*
- *notation*
- *redefine*
- *selector*
- *substitutionGroup*
- *unique*

13.1.1.1 Schema Package

The **Schema** icon in the XML Schema Toolbox creates an *XSDschema* stereotyped package that acts as a container for the XSD constructs, from which XML Schema can be generated.

XSD Properties: Project Browser: double-click on the *XSDschema* stereotyped Package
Diagram: double-click on the *XSDschema* stereotyped Package

Reference

Field/Button	Usage	See also
Schema Name	Specify the name for the Schema package.	
Target Namespace	Specify the namespace for this Schema package (<i>Optional</i>).	
Prefix	Specify the abbreviated value that represents the Target Namespace (<i>Optional</i>).	
Default Namespace	Specify the namespace for all non-prefixed <i>XSDelements</i> and	

	<i>XSDattributes (Optional).</i>																	
Schema File	Specify the file path where the XML Schema for this package is to be generated.																	
XMLNS	<p>List specifying the additional namespace and namespace-prefix pairs used in this Schema package.</p> <table border="1"> <thead> <tr> <th>Button</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td></td> <td>Press the New button to add an entry in the list. Double-click on an entry in the list to edit the values.</td> </tr> <tr> <th>Field</th> <th>Usage</th> </tr> <tr> <td>Prefix</td> <td>Specify the abbreviated value that represents t</td> </tr> <tr> <td>Namespace</td> <td>Specify the namespace.</td> </tr> <tr> <td>OK</td> <td>Save the values entered and close the Names</td> </tr> <tr> <td>Cancel</td> <td>Discard the values entered and close the Nam</td> </tr> <tr> <td>Help</td> <td>Display this Help topic.</td> </tr> </tbody> </table> <p>Click on the Delete button to remove an entry from the list.</p>	Button	Usage		Press the New button to add an entry in the list. Double-click on an entry in the list to edit the values.	Field	Usage	Prefix	Specify the abbreviated value that represents t	Namespace	Specify the namespace.	OK	Save the values entered and close the Names	Cancel	Discard the values entered and close the Nam	Help	Display this Help topic.	
Button	Usage																	
	Press the New button to add an entry in the list. Double-click on an entry in the list to edit the values.																	
Field	Usage																	
Prefix	Specify the abbreviated value that represents t																	
Namespace	Specify the namespace.																	
OK	Save the values entered and close the Names																	
Cancel	Discard the values entered and close the Nam																	
Help	Display this Help topic.																	
OK	Save the values entered and close the dialog.																	
Cancel	Discard the values entered and close the dialog.																	
Help	Display this Help topic.																	
UML	<p>Open the UML Properties dialog.</p> <p>This button is available when editing the Schema package.</p>	Properties <small>843</small>																

13.1.1.2 Global Element

The **Element** icon from the XML Schema Toolbox creates an *XSDtopLevelElement* stereotyped Class, which acts as a XSD global element, when dropped directly on a diagram.

XSD Properties: Project Browser: double-click on the *XSDtopLevelElement* stereotyped Class
Diagram: double-click on the *XSDtopLevelElement* stereotyped Class

Reference

Field/Button	Usage	See also						
Name	Specify the name for this global element.							
Type	Specify either an XSD <i>built-in dataType</i> or <i>complexType</i> or <i>simpleType</i> .							
Value	<p>If Type is an XSD <i>built-in type</i> then optionally specify its value:</p> <table border="1"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Default</td> <td>Specifies Value as default value.</td> </tr> <tr> <td>Fixed</td> <td>Specifies Value as fixed.</td> </tr> </tbody> </table>	Option	Usage	Default	Specifies Value as default value.	Fixed	Specifies Value as fixed.	
Option	Usage							
Default	Specifies Value as default value.							
Fixed	Specifies Value as fixed.							
Nested complexType	Check this option to create an <i>XSDcomplexType</i> as a child of this global element.	Complex Type ^[2037]						
Nested simpleType	Check this option to create an <i>XSDsimpleType</i> as a child of this global element.	Simple Type ^[2041]						
Annotation	Specify the notes for this element (<i>Optional</i>).							
OK	Save the values entered and close the dialog.							
Cancel	Discard the values entered and close the dialog.							
Help	Display this Help topic.							
UML	Open the UML Properties dialog. This button is available when editing the global element.	Properties ^[843]						

--	--	--

Notes

- The fields **Nested complexType** and **Nested simpleType** are available in the dialog only when creating a new global element (and not when editing the global element)
- The fields **Type**, **Nested complexType** and **Nested simpleType** are mutually exclusive - selecting one disables the others
- Global elements cannot contain any UML Attributes
- Global elements cannot be the source of Association connector
- Global elements cannot be the target of a Generalization connector
- Global elements can be the target of Association connector from Complex Type or Group Class
- Global elements can have at most one Generalization connector to a Complex Type or Simple Type Class

13.1.1.3 Local Element

The **Element** icon from the XML Schema Toolbox creates an *XSDelement* stereotyped UML Attribute, which acts as a local XSD element, when dropped on a *XSDcomplexType* and *XSDgroup* stereotyped classes.

XSD Properties: Project Browser: double-click on the *XSDelement* stereotyped Class
Diagram: double-click on the *XSDelement* stereotyped Class

Reference

Field/Button	Usage	See also
Name	Specify the name for this local element.	
Type	Specify either an XSD <i>built-in dataType</i> or <i>complexType</i> or <i>simpleType</i> .	Complex Type ^[2037] Simple Type ^[2041]
Reference	Specify whether to use the <i>ref</i> attribute (instead of the <i>type</i> attribute) to refer to the <i>complexType</i> or <i>simpleType</i> selected in the Type field in the generated XSD (<i>Optional</i>).	
Value	If Type is an XSD <i>built-in type</i> then optionally specify its value.	
	Option	Usage
	Default	Specifies Value as default value.
	Fixed	Specifies Value as fixed.

MinOccurs	Specify the minimum number of times this element occurs in the Class (<i>Optional</i>).							
MaxOccurs	Specify the maximum number of times this element occurs in the Class (<i>Optional</i>).							
Form	Specify whether to qualify the element (<i>Optional</i>): <table border="1" data-bbox="501 566 1235 860"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>qualified</td> <td>Use the Prefix supplied on the Schema package to qualify this element.</td> </tr> <tr> <td>unqualified</td> <td>Do not qualify this element.</td> </tr> </tbody> </table>	Option	Usage	qualified	Use the Prefix supplied on the Schema package to qualify this element.	unqualified	Do not qualify this element.	Schema Package ^[2029]
Option	Usage							
qualified	Use the Prefix supplied on the Schema package to qualify this element.							
unqualified	Do not qualify this element.							
Annotation	Specify the notes for this local element (<i>Optional</i>).							
OK	Save the values entered and close the dialog.							
Cancel	Discard the values entered and close the dialog.							
Help	Display this Help topic.							
UML	Open the UML Properties dialog. This button is available when editing the local element.	Properties ^[887]						

Notes

- Only Complex Types, Groups and Model Groups can have this UML Attribute

13.1.1.4 Global Attribute

The **Attribute** icon from the XML Schema Toolbox creates an *XSDtopLevelAttribute* class, which acts as an XSD global attribute, when dropped directly on a diagram.

XSD Properties:

Double-click on the *XSDtopLevelAttribute* stereotyped Class in the diagram or the Project Browser

Reference

Field/Button	Usage	See also						
Name	Specify the name for this global attribute.							
Type	Specify either an XSD <i>built-in dataType</i> or <i>simpleType</i> .							
Value	<p>If Type is an XSD <i>built-in type</i> then optionally specify its value.</p> <table border="1"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Default</td> <td>Specifies Value as default value.</td> </tr> <tr> <td>Fixed</td> <td>Specifies Value as fixed.</td> </tr> </tbody> </table>	Option	Usage	Default	Specifies Value as default value.	Fixed	Specifies Value as fixed.	
Option	Usage							
Default	Specifies Value as default value.							
Fixed	Specifies Value as fixed.							
Nested simpleType	Check this option to create an <i>XSDsimpleType</i> as a child of this global attribute.	Simple Type ^[204]						
Form	<p>Specify whether to qualify the attribute (<i>Optional</i>).</p> <table border="1"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>qualified</td> <td>Use the Prefix supplied on the Schema package to this attribute.</td> </tr> <tr> <td>unqualified</td> <td>Do not qualify this attribute.</td> </tr> </tbody> </table>	Option	Usage	qualified	Use the Prefix supplied on the Schema package to this attribute.	unqualified	Do not qualify this attribute.	Schema Package ^[2029]
Option	Usage							
qualified	Use the Prefix supplied on the Schema package to this attribute.							
unqualified	Do not qualify this attribute.							
Annotation	Specify the notes for this attribute (<i>Optional</i>).							
OK	Save the values entered and close the dialog.							
Cancel	Discard the values entered and close the dialog.							
Help	Display this Help topic.							
UML	<p>Open the UML Properties dialog.</p> <p>This button is available when editing the global attribute.</p>	Properties ^[843]						

Notes

- The field **Nested simpleType** is available in the dialog only when creating a new global attribute (and not when editing the global attribute)
- The fields **Type** and **Nested simpleType** are mutually exclusive - selecting one disables the other
- Global attributes cannot contain any UML Attributes
- Global attributes cannot be the source of Association connector
- Global attributes cannot be the target of a Generalization connector
- Global attributes can be the target of Association connector from Complex Type Classes
- Global attributes can have at most one Generalization connector to a Simple Type Class

13.1.1.5 Local Attribute

The **Attribute** icon from the XML Schema Toolbox creates an *XSDattribute* stereotyped UML Attribute, which acts as a local XSD attribute, when dropped on *XSDcomplexType* and *XSDattributeGroup* stereotyped Classes.

XSD Properties:

Double-click on the *XSDattribute* stereotyped UML Attribute in the diagram or the Project Browser

Reference

Field/Button	Usage	See also						
Name	Specify the name for this local attribute.							
Type	Specify either an XSD <i>built-in dataType</i> or <i>simpleType</i> .	Simple Type ^[204]						
Reference	Specify whether to use <i>ref</i> attribute (instead of <i>type</i> attribute) to refer to the <i>simpleType</i> selected in the Type field in the generated XSD (<i>Optional</i>).							
Value	<p>If Type is an XSD <i>built-in type</i> then optionally specify its value (<i>Optional</i>).</p> <table border="1"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Default</td> <td>Specifies Value as default value.</td> </tr> <tr> <td>Fixed</td> <td>Specifies Value as fixed.</td> </tr> </tbody> </table>	Option	Usage	Default	Specifies Value as default value.	Fixed	Specifies Value as fixed.	
Option	Usage							
Default	Specifies Value as default value.							
Fixed	Specifies Value as fixed.							
Form	Specify whether to qualify the attribute (<i>Optional</i>):							

	Option	Usage	
	qualified	Use the Prefix supplied on the Schema package to this attribute.	Schema Package ^[2023]
	unqualified	Do not qualify this attribute.	
Annotation	Specify the notes for this local attribute (<i>Optional</i>).		
OK	Save the values entered and close the dialog.		
Cancel	Discard the values entered and close the dialog.		
Help	Display this Help topic.		
UML	Open the UML Properties dialog. This button is available when editing the local attribute.		Properties ^[887]

Notes

- Only Complex Types and Attribute Groups can have this UML Attribute

13.1.1.6 Attribute Group

The **Attribute Group** icon from the XML Schema Toolbox creates an *XSDattributeGroup* class when dropped on a diagram. This Class is used to group a set of *XSDattribute* stereotyped UML Attributes and Simple Type Classes that can be referenced from an *XSDcomplexType* Class.

XSD Properties:

Double-click on the *XSDattributeGroup* stereotyped Class in the diagram or the Project Browser

Reference

Field/Button	Usage	See also
Name	Specify the name for this attribute group.	
Annotation	Specify the notes for this attribute (<i>Optional</i>).	

OK	Save the values entered and close the dialog.	
Cancel	Discard the values entered and close the dialog.	
Help	Display this Help topic.	
UML	Open the UML Properties dialog. This button is available when editing the attribute group.	Properties [843]

Notes

- Attribute Group cannot be the child of any other XSD class
- Attribute Group can contain only *XSDattribute* stereotyped UML Attributes and Simple Type classes
- Attribute Group can be the source of Association connector to another Attribute Group
- Attribute Group can be the target of Association connector from Complex Type classes
- Attribute Group cannot be the source or target of a Generalization connector

13.1.1.7 Complex Type

The **Complex Type** icon from the XML Schema Toolbox creates an *XSDcomplexType* stereotyped Class when dropped on a diagram.

XSD Properties:

Double-click on the *XSDcomplexType* stereotyped Class in the diagram or the Project Browser

Reference

Field/Button	Usage	See also						
Name	Specify the name for this complexType.							
Model Group	Specify how the child elements of this complexType should occur in the Schema.							
	<table border="1"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>sequence</td> <td>Specifies that the child elements must occur in the specified order.</td> </tr> <tr> <td>choice</td> <td>Specifies that only one of the child elements must occur.</td> </tr> </tbody> </table>	Option	Usage	sequence	Specifies that the child elements must occur in the specified order.	choice	Specifies that only one of the child elements must occur.	
	Option	Usage						
sequence	Specifies that the child elements must occur in the specified order.							
choice	Specifies that only one of the child elements must occur.							

	all	Specifies that the child elements can occur in any order	
MinOccurs		Specify the minimum number of times this element occurs in the Class (<i>Optional</i>).	
MaxOccurs		Specify the maximum number of times this element occurs in the Class (<i>Optional</i>).	
Annotation		Specify the notes for this element (<i>Optional</i>).	
Abstract		Specify whether this complexType can be used in the instance XML file (<i>Optional</i>). Default: False.	
Mixed		Specify whether character data can appear among the child elements (<i>Optional</i>). Default: False.	
OK		Save the values entered and close the dialog.	
Cancel		Discard the values entered and close the dialog.	
Help		Display this Help topic.	
UML		Open the UML Properties dialog. This button is available when editing the complexType.	Properties 843

Notes

- Complex Types can contain both *XSDelement* and *XSDattribute* stereotyped UML Attributes
- Complex Types can contain other Complex Types as child elements
- Complex Types can be children of Global Elements
- Complex Types can have Association connectors to other Complex Type, Simple Type, Attribute Group, Group and Model Group
- Complex Types can have at most one Generalization connector to either another Complex Type or a Simple Type Class

13.1.1.8 Model Group

The **Model Group** icon from the XML Schema Toolbox creates an *XSDsequence*, *XSDchoice* or *XSDall* stereotyped Class when dropped on a diagram.

XSD Properties:

Double-click on the Model Group in the diagram or the Project Browser

Reference

Field/Button	Usage	See also								
Name	Specify the name for this Model Group.									
Model Group	Specify how the child elements should occur. <table border="1" data-bbox="497 898 1284 1435"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>sequence</td> <td>This option creates an <i>XSDsequence</i> stereotyped Class. It specifies that the child elements must occur in the specified order.</td> </tr> <tr> <td>choice</td> <td>This option creates an <i>XSDchoice</i> stereotyped Class. It specifies that only one of the child elements must occur.</td> </tr> <tr> <td>all</td> <td>This option creates an <i>XSDall</i> stereotyped Class. It specifies that the child elements can occur in any order.</td> </tr> </tbody> </table>	Option	Usage	sequence	This option creates an <i>XSDsequence</i> stereotyped Class. It specifies that the child elements must occur in the specified order.	choice	This option creates an <i>XSDchoice</i> stereotyped Class. It specifies that only one of the child elements must occur.	all	This option creates an <i>XSDall</i> stereotyped Class. It specifies that the child elements can occur in any order.	
Option	Usage									
sequence	This option creates an <i>XSDsequence</i> stereotyped Class. It specifies that the child elements must occur in the specified order.									
choice	This option creates an <i>XSDchoice</i> stereotyped Class. It specifies that only one of the child elements must occur.									
all	This option creates an <i>XSDall</i> stereotyped Class. It specifies that the child elements can occur in any order.									
MinOccurs	Specify the minimum number of times this Model Group occurs in the Class (<i>Optional</i>).									
MaxOccurs	Specify the maximum number of times this Model Group occurs in the Class (<i>Optional</i>).									
Annotation	Specify the notes for this Model Group (<i>Optional</i>).									
OK	Save the values entered and close the dialog.									
Cancel	Discard the values entered and close the dialog.									

Help	Display this Help topic.	
UML	Open the UML Properties dialog. This button is available when editing the Model Group.	Properties [843]

Notes

- Model Groups can contain only XSDelement stereotyped UML Attributes
- Model Groups can contain other Complex Types and Simple Types as child elements
- Model Groups can have Association connector to other Complex Type, Simple Type, Group and Model Group
- Model Groups must have at least one incoming Association connector from a Complex Type
- Model Groups cannot be the source or target of Generalization connectors

13.1.1.9 Group

The **Group** icon from the XML Schema Toolbox creates an *XSDgroup* stereotyped Class when dropped on a diagram. This Class is used to group a set of *XSDelement* stereotyped UML Attributes, **Complex Type** and **Simple Type** Classes that can be referenced from an *XSDcomplexType* Class.

XSD Properties:

Double-click on the *XSDgroup* stereotyped Class in the diagram or the Project Browser.

Reference

Field/Button	Usage	See also								
Name	Specify the name for this group.									
Model Group	Specify how the child elements of this group should occur in the Complex Type: <table border="1" data-bbox="507 1579 1268 1973"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>sequence</td> <td>Specifies that the child elements must occur in the specified order.</td> </tr> <tr> <td>choice</td> <td>Specifies that only one of the child elements must occur.</td> </tr> <tr> <td>all</td> <td>Specifies that the child elements can occur in any order.</td> </tr> </tbody> </table>	Option	Usage	sequence	Specifies that the child elements must occur in the specified order.	choice	Specifies that only one of the child elements must occur.	all	Specifies that the child elements can occur in any order.	Complex Type [2037]
Option	Usage									
sequence	Specifies that the child elements must occur in the specified order.									
choice	Specifies that only one of the child elements must occur.									
all	Specifies that the child elements can occur in any order.									

Annotation	Specify the notes for this group (<i>Optional</i>).	
OK	Save the values entered and close the dialog.	
Cancel	Discard the values entered and close the dialog.	
Help	Display this Help topic.	
UML	Open the UML Properties dialog. This button is available when editing the group.	Properties 8431

Notes

- Group can contain only *XSDelement* stereotyped UML Attributes
- Group can contain Complex Types and Simple Types as child elements
- Group can have Association connector to other Complex Types, Simple Types and Groups
- Group can be the target of Association connector from Complex Types
- Group cannot be the source or target of a Generalization connector

13.1.1.10 Simple Type

The **Simple Type** icon from the XML Schema Toolbox creates an *XSDsimpleType* class when dropped on a diagram.

XSD Properties:

Double-click on the *XSDsimpleType* stereotyped Class in the diagram or the Project Browser

Reference

Field/Button	Usage	See also				
Name	Specify the name for this simpleType.					
Type	Specify either an XSD <i>built-in dataType</i> or simpleType: <table border="1" data-bbox="491 1771 1276 2007"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Restriction</td> <td>Restricts the value of this simpleType to the selected Type. The various restrictions (facets) on the simpleType are a tagged-values on this Class.</td> </tr> </tbody> </table>	Field	Usage	Restriction	Restricts the value of this simpleType to the selected Type . The various restrictions (facets) on the simpleType are a tagged-values on this Class.	
Field	Usage					
Restriction	Restricts the value of this simpleType to the selected Type . The various restrictions (facets) on the simpleType are a tagged-values on this Class.					

	List	Specifies this simpleType as a list of values of the select	
Annotation	Specify the notes for this simpleType (<i>Optional</i>).		
OK	Save the values entered and close the dialog.		
Cancel	Discard the values entered and close the dialog.		
Help	Display this Help topic.		
UML	Open the UML Properties dialog. This button is available when editing the simpleType.		Properties [843]

Notes

- Simple Types cannot contain any XSDelement or XSDattribute stereotyped UML Attributes
- Simple Types cannot contain any child classes
- Simple Types cannot be the source of an Association connector
- Simple Types can be the target of a Generalization connector
- Simple Types can have at most one Generalization connector to another Simple Type Class

13.1.1.11 Union

The **Union** icon from the XML Schema Toolbox creates an *XSDunion* class when dropped directly on a diagram. This Class, which is a Simple Type, defines a collection of Simple Types.

XSD Properties:

Double-click on the *XSDunion* stereotyped Class in the diagram or the Project Browser

Reference

Field/Button	Usage	See also
Name	Specify the name for this Union.	
Member Types	Specify the collection of XSD <i>built-in dataTypes</i> and simpleTypes.	Simple Type [2047]
Annotation	Specify the notes for this Union (<i>Optional</i>).	

OK	Save the values entered and close the dialog.	
Cancel	Discard the values entered and close the dialog.	
Help	Display this Help topic.	
UML	Open the UML Properties dialog. This button is available when editing the Union.	Properties [843]

Notes

- Unions cannot contain any child Classes
- Unions cannot contain any XSDelement or XSDattribute stereotyped UML Attributes
- Unions cannot be the source of an Association connector
- Unions can be the target of Association connector from Complex Types
- Unions can be the target of a Generalization connector from Simple Types

13.1.1.12 Enumeration

The **Enum** icon from the XML Schema Toolbox creates an *enumeration* stereotyped Class when dropped on a diagram. This Class, which is a Simple Type, defines a list of acceptable values for the Class.

XSD Properties:

Double-click on the *enumeration* stereotyped Class under a Schema package in the diagram or the Project Browser

Reference

Field/Button	Usage	See also
Name	Specify the name for this Enumeration.	
Type	Specify either an XSD <i>built-in dataType</i> or simpleType.	Simple Type [2041]
Values	Specify the list of acceptable values for the specified Type .	
Annotation	Specify the notes for this Enumeration (<i>Optional</i>).	

OK	Save the values entered and close the dialog.	
Cancel	Discard the values entered and close the dialog.	
Help	Display this Help topic.	
UML	Open the UML Properties dialog. This button is available when editing the Enumeration.	Properties ⁸⁴³

Notes

- Enumeration cannot contain any *XSDelement* or *XSDattribute* stereotyped UML Attributes
- Enumeration cannot contain any child Classes
- Enumeration cannot be the source of an Association connector
- Enumeration can be the target of a Generalization connector
- Enumeration can have at most one Generalization connector to another Simple Type Class

13.1.1.13 Any


The **Any** icon from the XML Schema Toolbox creates an *XSDany* stereotyped Class when dropped on a diagram. This Class allows a Complex Type to contain elements that are not specified in the Schema package.

XSD Properties:

Double-click on the *XSDany* stereotyped Class in the diagram or the Project Browser

Reference

Field/Button	Usage	See also
Name	Specify the name for this Class.	
Namespace	Specifies the namespace that contains the elements that can be used in the Complex Type (<i>Optional</i>).	Complex Type ²⁰³⁷
ProcessContents	Specifies how the XML Parser should validate these elements (<i>Optional</i>):	
	Option	Usage
	lax	Tries to validate the elements against their Schema is flagged when the Schema cannot be obtained.

	<table border="1"> <tr> <td></td> <td></td> </tr> <tr> <td>skip</td> <td>Skips validating the elements.</td> </tr> <tr> <td>strict</td> <td>Validates the elements against their Schema.</td> </tr> </table>			skip	Skips validating the elements.	strict	Validates the elements against their Schema.	
skip	Skips validating the elements.							
strict	Validates the elements against their Schema.							
Annotation	Specify the notes for this Class (<i>Optional</i>).							
OK	Save the values entered and close the dialog.							
Cancel	Discard the values entered and close the dialog.							
Help	Display this Help topic.							
UML	Open the UML Properties dialog. This button is available when editing this Class.	Properties 						

Notes

- Any Class cannot contain any UML Attributes or child XSD Classes
- Any Class cannot be the child of any XSD Class
- Any Class cannot be the target of Generalization connectors
- Any Class cannot be the source of Association and Generalization connectors
- Any Class can be the target of Association connectors from Complex Types, Groups and Model Groups
- Any Class must have at least one incoming Association connector

13.1.1.14 Any Attribute

The **Any Attribute** icon from the XML Schema Toolbox creates an *XSDany* stereotyped UML Attribute when dropped on Complex Type or an Attribute Group. This Class allows a Complex Type or an Attribute Group to contain elements that are not specified in the Schema package.

XSD Properties:

Double-click on the *XSDany* stereotyped UML Attribute in a Class or the Project Browser

Reference

Field/Button	Usage	See also								
Name	Specify the name for this UML Attribute.									
Namespace	Specifies the namespace that contains the attributes that can be used in the Complex Type or Attribute Group (<i>Optional</i>).	Complex Type <small>2037</small> Attribute Group <small>2036</small>								
ProcessContents	Specifies how the XML Parser should validate these attributes (<i>Optional</i>): <table border="1" data-bbox="496 629 1227 1025"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>lax</td> <td>Tries to validate the attribute against their Schema; r is flagged when the Schema cannot be obtained.</td> </tr> <tr> <td>skip</td> <td>Skips validating the attributes.</td> </tr> <tr> <td>strict</td> <td>Validates the attributes against their Schema.</td> </tr> </tbody> </table>	Option	Usage	lax	Tries to validate the attribute against their Schema; r is flagged when the Schema cannot be obtained.	skip	Skips validating the attributes.	strict	Validates the attributes against their Schema.	
Option	Usage									
lax	Tries to validate the attribute against their Schema; r is flagged when the Schema cannot be obtained.									
skip	Skips validating the attributes.									
strict	Validates the attributes against their Schema.									
Annotation	Specify the notes for this UML Attribute (<i>Optional</i>).									
OK	Save the values entered and close the dialog.									
Cancel	Discard the values entered and close the dialog.									
Help	Display this Help topic.									
UML	Open the UML Properties dialog. This button is available when editing this UML Attribute.	Properties <small>887</small>								

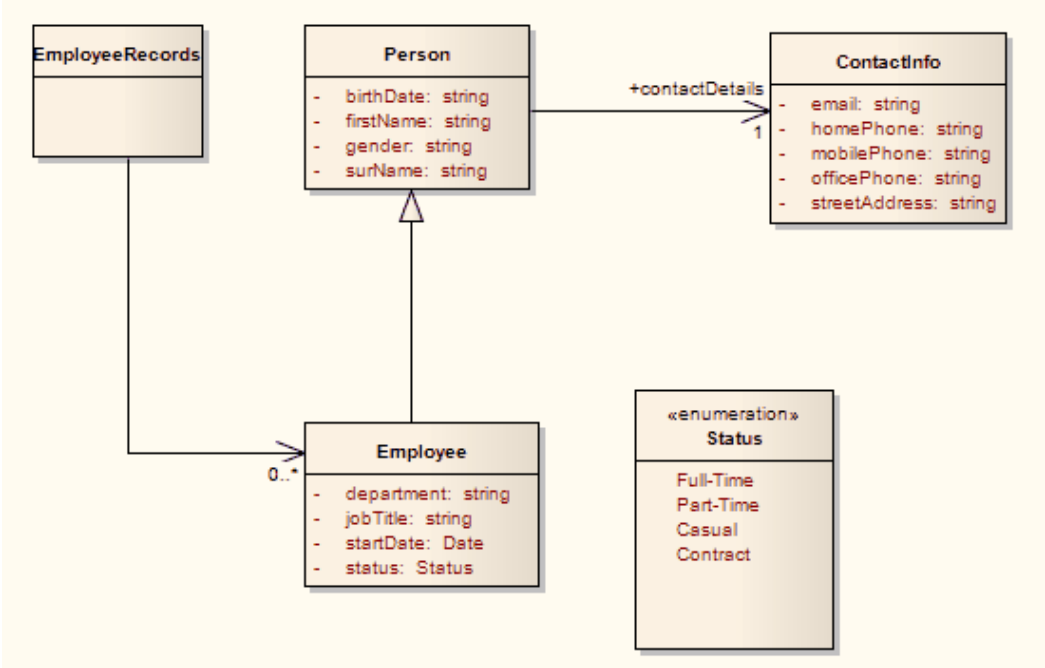
Notes

- Only Complex Types and Attribute Groups can have this UML Attribute

13.1.2 Abstract XSD models

XML schemas can also be modeled using simple, abstract Class models. This can be useful in enabling an architect to start working at a higher level of abstraction, without concern for the implementation details of a Schema. Such an abstract model can be refined further using the XML Schema pages of the Toolbox, or it can be generated directly by Enterprise Architect's **Schema Generator**. In this case, a set of **default mappings** is assumed by the schema generator to convert the abstract model to an XSD file.

Example

Topic	Detail
Diagram	<p>The following is a simplified version of the <i>Employee Details</i> example model, which does not use XSD-specific stereotypes or Tagged Values:</p>  <pre> classDiagram class EmployeeRecords class Person { - birthDate: string - firstName: string - gender: string - surName: string } class Employee { - department: string - jobTitle: string - startDate: Date - status: Status } class ContactInfo { - email: string - homePhone: string - mobilePhone: string - officePhone: string - streetAddress: string } class Status { <<enumeration>> Full-Time Part-Time Casual Contract } EmployeeRecords "0..*" --> Employee Employee < -- Person Person "1" --> ContactInfo : +contactDetails </pre>
Schema	<p>The following schema fragment would be generated by Enterprise Architect, given the above model:</p> <pre> <?xml version="1.0" ?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" > <xs:simpleType name="Status" > <xs:restriction base="xs:string" > <xs:enumeration value="Full-Time" /> <xs:enumeration value="Part-Time" /> <xs:enumeration value="Casual" /> <xs:enumeration value="Contract" /> </xs:restriction> </xs:simpleType> <xs:element name="Person" type="Person" /> <xs:complexType name="Person" > <xs:sequence> <xs:element name="firstName" type="xs:string" /> <xs:element name="surName" type="xs:string" /> <xs:element name="birthDate" type="xs:string" /> <xs:element name="gender" type="xs:string" /> <xs:element name="contactDetails" type="ContactInfo" /> </xs:sequence> </xs:complexType> <xs:element name="Employee" type="Employee" /> <xs:complexType name="Employee" > <xs:complexContent> </pre>

Topic	Detail
	<pre> <xs:extension base="Person"> <xs:sequence> <xs:element name="status" type="Status"/> <xs:element name="jobTitle" type="xs:string"/> <xs:element name="startDate" type="xs:date"/> <xs:element name="department" type="xs:string"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> <xs:element name="EmployeeRecords" type="EmployeeRecords"/> <xs:complexType name="EmployeeRecords"> <xs:sequence> <xs:element name="Employee" type="Employee" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> <xs:element name="ContactInfo" type="ContactInfo"/> <xs:complexType name="ContactInfo"> <xs:sequence> <xs:element name="homePhone" type="xs:string"/> <xs:element name="mobilePhone" type="xs:string"/> > <xs:element name="officePhone" type="xs:string"/> > <xs:element name="email" type="xs:string"/> <xs:element name="streetAddress" type="xs: string"/> </xs:sequence> </xs:complexType> </xs:schema> </pre>

Learn more

- [Default UML to XML Mappings](#) ^[2048]
- [Generate XSD](#) ^[2049]

13.1.2.1 Default UML to XSD Mappings

The following table describes the default mapping of UML to XSD constructs. This set of mappings is useful when defining simple schemas from abstract Class models. The defaults are also assumed by the schema generator when generating unsteretyped elements in an abstract model. The XML Schema pages of the Toolbox (and UML Profile for XSD) override these default mappings through the use of stereotypes and Tagged Values.

Topics

UML Construct	Default XSD Production Rules
Package	<p>A schema element is generated for the target package. If the target package includes Classes from another package, which has the Tagged Values <i>targetNamespace</i> and <i>targetNamespacePrefix</i> set, these are included as attributes of the schema element.</p> <p>In addition, an <i>import</i> or <i>include</i> element is created for each referenced package. (An</p>

UML Construct	Default XSD Production Rules
	<i>include</i> element is used if the external package shares the same <i>targetNamespace</i> Tagged Value as the target package. An <i>import</i> element is used where the <i>targetNamespaces</i> differ).
Class	A root-level element declaration and <i>complexType</i> definition are generated. The element name and type are the same as the Class name. An XSD sequence model group is generated to contain UML attributes generated as elements.
Attribute	An element is declared for each Class attribute. The element name is set to that of the UML attribute name. This is prefixed with the Class name to make the element unique. The <i>minOccurs</i> and <i>maxOccurs</i> attributes are set to reflect the attribute cardinality. If the attribute refers to another Class, the element declaration is followed a <i>complexType</i> definition, which contains a reference to the appropriate <i>complexType</i> .
Association	An element is declared for each association owned by a Class. The element name is set to that of the association role. The <i>minOccurs</i> and <i>maxOccurs</i> attributes reflect the cardinality of the association.
Generalization (Inheritance)	For single inheritances, an extension element is generated with the base attribute set to the base Classname. The UML attributes of the child Class are then appended to an all model group within the extension element.
«enumeration» (stereotype)	A <i>simpleType</i> element is declared for the enumeration Class with the name attribute set to the Classname. A restriction element is generated with base set to string. Each of the Class attributes is appended to the restriction element as XSD enumeration elements with value set to the UML attribute name. Any type specification for the UML attributes is ignored by the schema generator.

Notes

- If left unspecified, *minOccurs* and *maxOccurs* default to 1
- If the direction of the Association is unspecified, the owner is assumed to be the source

13.1.3 Generate XSD

The **Generate XML Schema** feature forward-engineers a UML Class model to a W3C XML Schema (XSD) file. An XML schema corresponds to a UML package in Enterprise Architect, therefore XML schema generation is a package-level operation.

Before you begin this procedure, check that the location of the file into which the XML Schema is to be generated, has been defined in the Schema Package (see the *Schema Package* topic).

Access [Tools | XML Schema | Generate XML Schema](#), or
[Project Browser package context menu | Code Engineering | Generate XML Schema](#)

Reference

Field/Option/ Button	Usage	See also	
Encoding	Specify the required XML encoding.		
XSD Style	Field	Generate Global Element ^[2051]	
	Generate global element for all global ComplexTypes ('Garden of Eden' style)		Usage Generate Schema in the Gal This option is checked, by de
Referenced Package Options	Field	Schema Package ^[2029]	
	Generate XSD for Referenced packages		Usage Generate Schema for packages referenced by any of the packages in the list box.
	Prompt when missing Filename		Usage Enable Enterprise Architect to prompt for filename for a referenced package during Schema generation, if the package name generate the Schema file is missing.
Use relative path to reference XSDs (if 'schemaLocation' tag is empty)	Usage Use a relative-path in the XSD <i>include</i> statement when referencing packages, provided that the path is empty on the referenced package. Ensure that the Schema File XSD Schema Properties dialog Properties dialog for a Schema referenced and referencing packages, for the stereotyped packages, for the correctly determined.		
Child Package Options	Field		
	Generate XSD for Child Packages		Usage Generate schema for child packages selected package. <ul style="list-style-type: none"> • Include all packages packages under the the list box • Include <XSDschem

Field/Option/ Button	Usage	See also
	<div style="border: 1px solid black; width: 100%; height: 100%;"></div> <p>only those packages stereotype «XSDsch</p> <p>The list-box lists, for each package, the package name and the file path where the Schema file is to be generated.</p> <p>Change the file path for a package by double-clicking on the entry in the list-box, if required.</p> <p>Ensure that the checkbox is selected for each package required for generation.</p>	
Generate	Generate the Schema for each of the packages checked in the list-box.	
Close	Close this dialog.	
Help	Display this Help topic.	
View Schema	View the generated Schema for a package highlighted in the list-box.	
Progress	Lists the progress of Schema Generation.	

Learning Center topics

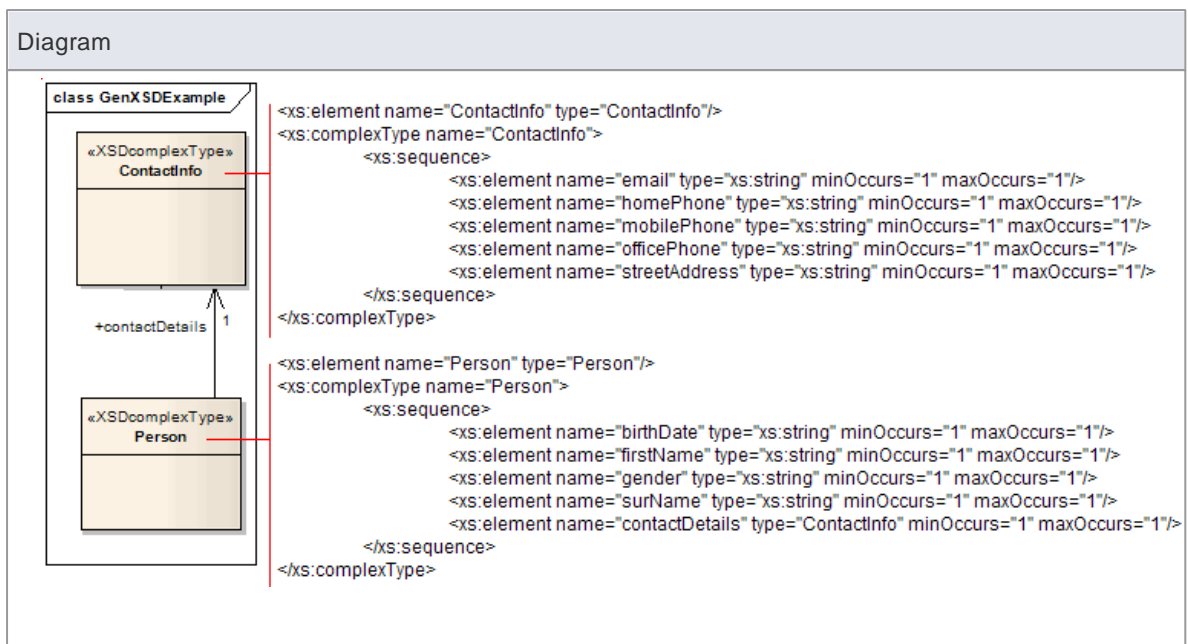
- (Alt+F1) | [SOA and XML Engineering | XSD | Generate XML Schema](#)

13.1.3.1 Generate Global Element

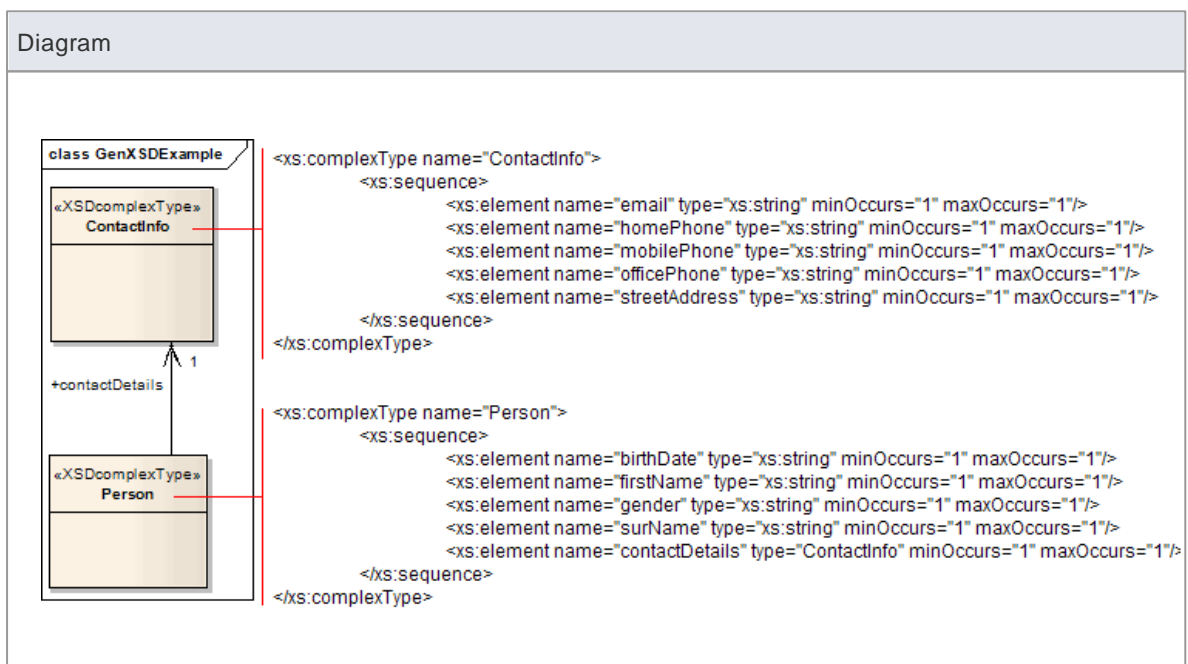
Enterprise Architect, by default, generates XML Schema in the *Garden of Eden* style. For every global *XSDcomplexType* stereotyped Class, Enterprise Architect generates a global element.

Example

Diagram



You can change the above specified default behaviour by de-selecting the **Generate global element for all global ComplexTypes** checkbox on the Generate XML Schema dialog. Then, the generated XSD no longer contains the global element.



Learn more

- [Generate XML Schema](#) 2049

13.1.4 Import XSD

The **XML Schema Import** facility is used to reverse engineer a W3C XML Schema (XSD) file as a UML Class model. An XSD file is imported into Enterprise Architect as a UML package.

Access

- **Tools | XML Schema | Import XML Schema**
- **Project Browser package context menu | Code Engineering | Import XML Schema**

Reference

Field/Option/Button	Usage	See also										
Root Package	Indicates the name of the package selected.											
Directory	Specifies the directory where the XSD file(s) is located.											
Selected File(s)	Lists the XML Schema(s) currently selected for import. <ul style="list-style-type: none"> • To select several individual files (Ctrl) + click on each file. • To select a range of files, press (Shift) and select the first and last file in the range. 											
Import Options	<table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Import global elements with "Type" postfix</td> <td>Treat the global element as a <i>ComplexType</i> if it is referenced by any of the Selected File(s) field.</td> </tr> <tr> <td>Import referenced XML Schema(s)</td> <td>Import any XML Schema referenced by any of the Selected File(s) field.</td> </tr> <tr> <td>Create Diagram for XML Schema(s)</td> <td>Creates a Class diagram for the imported <i>XSDschema</i>.</td> </tr> <tr> <td>Import XSD Elements/Attributes as</td> <td>Indicate how the inline <i>XSDattributes</i> are imported.</td> </tr> </tbody> </table>	Field	Usage	Import global elements with "Type" postfix	Treat the global element as a <i>ComplexType</i> if it is referenced by any of the Selected File(s) field.	Import referenced XML Schema(s)	Import any XML Schema referenced by any of the Selected File(s) field.	Create Diagram for XML Schema(s)	Creates a Class diagram for the imported <i>XSDschema</i> .	Import XSD Elements/Attributes as	Indicate how the inline <i>XSDattributes</i> are imported.	Global Element and ComplexType <small>[2054]</small>
	Field	Usage										
	Import global elements with "Type" postfix	Treat the global element as a <i>ComplexType</i> if it is referenced by any of the Selected File(s) field.										
	Import referenced XML Schema(s)	Import any XML Schema referenced by any of the Selected File(s) field.										
Create Diagram for XML Schema(s)	Creates a Class diagram for the imported <i>XSDschema</i> .											
Import XSD Elements/Attributes as	Indicate how the inline <i>XSDattributes</i> are imported.											
Import	Initiate XSD Import.											
Close	Close this dialog.											

Help	Display this Help topic	
Progress	Lists the progress of Schema Import.	

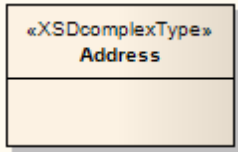
Notes

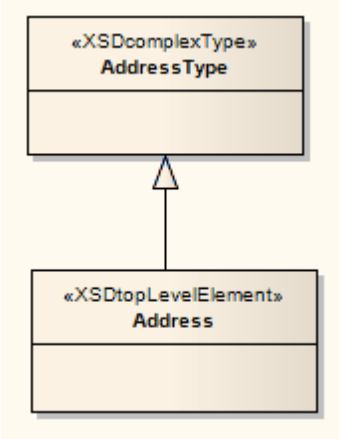
- If an XML Schema file being imported already exists in the model, Enterprise Architect skips importing the file
- References to XSD *Primitive Types* are always imported as UML Attributes
- References to XSD constructs in external Schema files are always imported as UML Attributes
- Enterprise Architect uses the *schemaLocation* attribute in the *XSD Import* and *XSD Include* elements of an XML Schema to determine the dependencies between the files; ensure that this attribute is set to a valid file path (and not a URL) for the dependent XML Schema(s) to be imported correctly

Learning Center topics

- (Alt+F1) | [SOA and XML Engineering | XSD | Import XML Schema](#)

13.1.4.1 Global Element and ComplexType

Topic	Detail
ComplexType element naming convention	<p>Some XML Schemas have ComplexType elements with the same name as the referring global elements, but with the suffix Type as shown below:</p> <pre><xs:element name="Address" type="AddressType" / > <xs:complexType name="AddressType" > <xs:sequence / > </xs:complexType></pre> <p>On XSD import, Enterprise Architect treats this global element and its bounding ComplexType as a single entity and creates a single XSDcomplexType stereotyped Class with the same name as the global element as shown below:</p> 
Changing the default behavior	<p>You can change the above specified default behaviour by selecting the Import global elements with "Type" postfix checkbox. When you select this option, Enterprise Architect treats the global element and the ComplexType it is referring to as two</p>

Topic	Detail
	<p>separate entities.</p> <p>So, for the above example, Enterprise Architect creates an <i>XSDtopLevelElement</i> stereotyped Class for the global element and an <i>XSDcomplexType</i> stereotyped Class for the ComplexType, and connects them as follows:</p>  <pre> classDiagram class AddressType["«XSDcomplexType» AddressType"] class Address["«XSDtopLevelElement» Address"] Address -- > AddressType </pre>

Notes

- Enterprise Architect treats the following as two separate entities irrespective of whether the **Import global elements with "Type" postfix** checkbox is selected or unselected:

```

<xs:element name="HomeAddress" type="AddressType" />
<xs:complexType name="AddressType">
  <xs:sequence />
</xs:complexType>
  
```

Learn more

- [Import XML Schema](#)^[2053]

13.1.5 UML Profile for XSD

The **UML Profile for XSD** specifies a set of stereotypes, tagged values and constraints that can be applied to the UML model in order to change particular aspects of the resulting schema. For example, you might have to convert certain UML Class attributes to XSD attributes, or use a model group other than the default *Sequence*.

Enterprise Architect provides native support for the UML Profile for XSD via the XML schema page of the Diagram Toolbox. Alternatively, you can use the profile via Enterprise Architect's generic profile mechanism by downloading the UML Profile for XSD. The XSD profile used by Enterprise Architect is an adaptation of the profile defined in *Modeling XML Applications with UML* (David Carlson).

Learn more

- [Profile for XML Schema](#)^[2056]
- [XSD Datatypes Package](#)^[2065]

13.1.5.1 Profile for XML Schema

The **UML Profile for XSD** specifies stereotypes that provide an explicit mapping from XSD to UML constructs. The Tagged Values further define aspects of the mapping, such as whether the elements should be qualified. The constraints define any conditions that must be satisfied for the stereotype to apply.


The UML Profile for XSD specifies the following stereotypes:

- [XSDschema](#)^[2056]
- [XSDcomplexType](#)^[2057]
- [XSDsimpleType](#)^[2058]
- [XSDsequence](#)^[2059]
- [XSDchoice](#)^[2060]
- [XSDelement](#)^[2060]
- [XSDattribute](#)^[2061]
- [XSDany](#)^[2062]
- [XSDrestriction](#)^[2062]
- [XSDgroup](#)^[2063]
- [XSDtopLevelElement](#)^[2063]
- [XSDtopLevelAttribute](#)^[2064]
- [XSDunion](#)^[2064]
- [XSDattributeGroup](#)^[2065]

The following tables list the features of the UML Profile for XSD.

- Tagged Value names are shown in bold followed by the allowed values
- If a default value is used by Enterprise Architect's schema generator, it is underlined

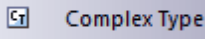
«XSDschema»:

UML Construct		Package.
Description		All Classes in a package are defined within one schema. This stereotype can be used to specify schema-wide settings.
Toolbox Icon		 Schema
Tagged Values	anonymousRole (true <u>false</u>)	Specifies if the role name is included in the element declaration for the UML attribute.

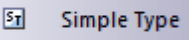
	anonymousType (true <u>false</u>)	Specifies whether the Class type is anonymous for attributes.
	attributeFormDefault (qualified <u>unqualified</u>)	Determines whether attribute instances must be qualified.
	defaultNamespace	The default namespace used in this schema. This value is used to specify the default namespace attribute (<i>xmlns=</i>), in the schema element.
	elementDerivation (<u>true</u> false)	Determines whether inheritances are generated using XSD extension or copy-down inheritance.
	elementFormDefault (qualified <u>unqualified</u>)	Determines whether element instances must be qualified.
	memberNames (<u>qualified</u> unqualified)	Determines whether elements generated from Class attributes have their name qualified by the corresponding Class name.
	modelGroup (all <u>sequence</u> choice)	Specifies the default XSD model group used to generate <i>complexType</i> definitions.
	schemaLocation	The URI that identifies the location of the schema. This value is used in the import and include elements.
	targetNamespace	The URI that uniquely identifies this schema's namespace.
	targetNamespacePrefix	The prefix that abbreviates the <i>targetNamespace</i> .
	version	The version of this schema.
	xmlns	The comma separated list of all the namespaces (and their prefixes).
Constraints		None.

«XSDcomplexType»:

UML		Class
-----	--	-------

Construct		
Description		<i>complexType</i> definitions are created for generic UML Classes. This stereotypes helps tailor the generation of a <i>complexType</i> definition.
Toolbox Icon		
Tagged Values	memberNames (qualified unqualified)	Determines whether elements generated from the UML Class attributes and associations have their name qualified by the corresponding Class name for this <i>complexType</i> definition.
	mixed (true <u>false</u>)	Determines whether this element can contain mixed element and character content. See the W3CXML Schema recommendation.
	modelGroup (all sequence choice)	Overrides the default XSD model for generating this <i>complexType</i> definition.
	minOccurs	Specifies the minimum number of times the model group in the element can occur.
	maxOccurs	Specifies the maximum number of times the model group in the element can occur.
	nilable (true <u>false</u>)	Specifies whether a null value can be assigned to the top-level element generated for this <i>complexType</i> element.
Constraints		None.

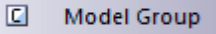
«XSDsimpleType»

UML Construct		Class
Description		An XSD <i>simpleType</i> is generated for Classes with this stereotype.
Toolbox Icon		

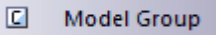
Tagged Values	derivation: (restriction list)	Specifies the derivation of the <i>simpleType</i> . See the W3C XML Schema recommendation.
	length:	See the W3C XML Schema recommendation.
	minLength:	
	maxLength:	
	minInclusive:	
	minExclusive:	
	maxInclusive:	
	maxExclusive:	
	totalDigits:	
	fractionDigits:	
	whiteSpace:	
	pattern:	
Constraints		This Class can only participate in an inheritance relation with another <i>simpleType</i> . It cannot have any attributes or own any associations; they are ignored if present.

«XSDsequence»

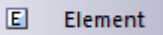
UML Construct		Class.
Description		<p>The schema generator creates a sequence model group as the container for the attributes and associations owned by this Class. The model group is in turn added to the model groups of this Class respective owners.</p> <p>Tagged values specified by owners of this Class persist</p>

		through to the child elements of this model group. Thus if <i>memberNames</i> are unqualified for a <i>complexType</i> , so are the children of this model group when added to that <i>complexType</i> .
Toolbox Icon		
Tagged Values		None.
Constraints		This Class must be the destination of unidirectional associations. If it is not, this Class and its connectors are ignored, possibly invalidating other model group Classes. Inheritance relations are ignored for this Class.

«XSDchoice»


UML Construct		Class.
Description		Creates an XSD choice element. See <i>XSDsequence</i> for more details.
Toolbox Icon		
Tagged Values		None.
Constraints		As for <i>XSDsequence</i> .

«XSDelement»

UML Construct		UML Attribute, Association <i>End</i> .
Description		By applying this stereotype to a UML Class or Association <i>End</i> , the corresponding UML entity is generated as an element within the parent <i>complexType</i> and not as an XSD attribute.
Toolbox Icon		

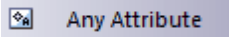
Tagged Values	form: (qualified unqualified)	Overrides the schema's <i>elementFormDefault</i> value.
	position:	Causes the elements to be ordered within a sequence model group of the containing <i>complexType</i> . Duplicated and invalid position Tagged Values are ignored and result in undefined ordering of the UML attributes. Missing position values cause the defined positions to be allocated as specified, with the remaining elements filling the missing positions in an undefined order.
	anonymousRole: (true <u>false</u>)	Specifies if the role name is included in the element declaration for the UML attribute.
	anonymousType: (true <u>false</u>)	Specifies whether the Class type is anonymous for attributes.
	nillable (true <u>false</u>)	Specifies whether a null value can be assigned to this element.
	default	See the W3C XML Schema recommendation.
	fixed	
Constraints		None.

«XSDattribute»

UML Construct		UML Attribute, Association <i>End</i> .
Description		By applying this stereotype to a UML Class attribute or Association <i>End</i> , the corresponding UML entity is generated as an XSD attribute within the parent <i>complexType</i> and not as an XSD element.
Toolbox Icon		
Tagged Values	form: (qualified unqualified)	Overrides the schema's <i>attributeFormDefault</i> value.
	use:	

	(prohibited <u>optional</u> required)	See the W3C XML Schema recommendation.
	default	
	fixed	
Constraints		The attribute <i>datatype</i> should not see a Class specification, otherwise it is ignored.

«XSDany»

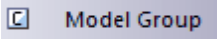
UML Construct		UML Attribute.
Description		If applied to a UML attribute, an XSD <i>anyAttribute</i> element is generated. If applied to a UML Class, an XSD <i>any</i> element is generated.
Toolbox Icon		 Any Attribute
Tagged Values	namespace:	See the W3C XML Schema recommendation.
	processContents: (skip lax <u>strict</u>)	
Constraints		None.

«XSDrestriction»

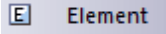
UML Construct		Generalization.
Description		Overrides the default use of XSD extension for inheritance and generates the child as a <i>complexType</i> with a restriction element instead.
Tagged Values		None.

Constraints		Applies only to UML Class parent-child relations.
-------------	--	---

«XSDgroup»


UML Construct		Class.
Description		An <i>XSDgroup</i> is generated for Classes with this stereotype.
Toolbox Icon		
Tagged Values	modelGroup: (<u>sequence</u> choice all)	Overrides the default XSD model for generating this group definition.
Constraints		A group Class can only associate itself to other group Classes. A group Class can be associated by another group Class or a <i>complexType</i> Class. The association should be via an Association connector. A group Class cannot be inherited/aggregated.

«XSDtopLevelElement»

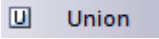
UML Construct		Class.
Description		Creates an <code><xs:element></code> construct which acts as a container for <i>XSDcomplexType</i> and <i>XSDsimpleType</i> Class.
Toolbox Icon		
Tagged Values	default	See the W3C XML Schema recommendation.
	fixed	
Constraints		An <i>XSDtopLevelElement</i> Class can contain either an <i>XSDsimpleType</i> or an <i>XSDcomplexType</i> as its child Class. When such a Class is present as its child, all its inheritance is ignored.

		This Class cannot be inherited.
--	--	---------------------------------

«XSDtopLevelAttribute»

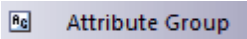
UML Construct		Class.
Description		Creates an <code><xs:attribute></code> construct which acts as a container for <i>XSDsimpleType</i> Class.
Toolbox Icon		
Tagged Values	use: (<u>optional</u> required prohibited)	See the W3C XML Schema recommendation.
	default	
	fixed	
Constraints		An <i>XSDtopLevelAttribute</i> Class can contain only an <i>XSDsimpleType</i> Class as its child Class. When such a Class is present as its child, all its inheritance is ignored. This Class can inherit from only one <i>XSDsimpleType</i> Class.

«XSDunion»

UML Construct		Class.
Description		Creates an <code><xs:union></code> construct which can act as a container for <i>XSDsimpleType</i> Class.
Toolbox Icon		
Tagged Values		None.

Constraints		<p>An <i>XSDunion</i> Class can contain only <i>XSDsimpleType</i> as its child Class and can generalize from other <i>XSDsimpleType</i> Classes only.</p> <p>All the Classes that this Class generalizes become the members of the attribute <i>memberTypes</i>.</p> <p>This Class cannot have any attributes or associations.</p>
-------------	--	--

«XSDattributeGroup»

UML Construct		Class.
Description		Creates an <code><XSDattributeGroup></code> construct which can act as a container for a set of elements for stereotype <i>XSDattribute</i> .
Toolbox Icon		
Tagged Values		None.
Constraints		<p>An <i>XSDattributeGroup</i> Class can contain only elements of stereotype <i>XSDattribute</i> and can be associated only with other <i>XSDattributeGroup</i> Classes.</p> <p>Only <i>XSDcomplexType</i> Classes can associate with this Class.</p> <p>This Class cannot be inherited.</p>

Learn more

- [UML Profile for XSD](#) (Online Resource)
- [W3C XML Schema](#) (Online Resource)
- [Using Profiles](#) ¹²⁷⁷

13.1.5.2 XSD Datatypes Package

When modeling XSD constructs, it is often useful to have the XSD primitive types represented as UML elements. In this way user-defined types, for example, can reference the datatype elements as part of inheritance or association relationships.

Sparx Systems provides the set of primitive XSD data types as a UML package in the form of an XMI file. Each of the XSD primitive types is represented by a UML Class in a package named *XSDDatatypes*.

How to

Step	Action	See also
1	Download the <i>XSDDataTypes</i> package using the link to the right. The file <i>XSDDataTypes.xml</i> is an XMI file.	XSDDataTypes Package (XML file)
2	Use Enterprise Architect's XMI import facility, which is available via the Project Model Import/Export Import Package from XMI menu option.	XMI import ^[41]
3	When the XMI import is complete, you have the UML package named <i>XSDDataTypes</i> in your model, from which you can drag and drop the relevant types as required.	

Notes

- Import and use the XSD Datatypes into Enterprise Architect only if you are modeling the XSD model using the UML Profile for XSD (instead of using the XML Schema Toolbox)

13.2 Web Services - WSDL

Enterprise Architect enables rapid modeling, forward engineering and reverse engineering of **Web Service Definition Language 1.1** (WSDL), a key W3C XML technology.

WSDL support is critical for the development of a complete *Service Oriented Architecture* (SOA), and the coupling of UML 2.3 and XML provides the natural mechanism for specifying, constructing and deploying XML-based SOA artifacts within an organization. This section explains how to use Enterprise Architect to model WSDL files.

Learn more

- [W3C WSDL Specification](#) (Online Resource)
- [Create WSDL 1.1 Model Structure](#)^[2067]
- [WSDL Model Structure](#)^[2070]
- [Model WSDL](#)^[2069]
- [WSDL Model Transformation](#)^[1705]
- [Generate WSDL 1.1](#)^[2085]
- [Import WSDL 1.1](#)^[2086]

Learning Center topics

- (Alt+F1) | [SOA and XML Engineering | WSDL](#)

13.2.1 Create WSDL 1.1 Model Structure

Enterprise Architect provides the WSDL page in the Toolbox to conveniently model WSDL documents.

WSDLs, represented as Components marked with the stereotype *WSDL*, are contained in a *WSDLnamespace* stereotyped Package whose hierarchy represents the target WSDL namespace and its constituent *XSD Types*, *Messages*, *PortTypes*, *Bindings* and *Services*.

A sample WSDL Package structure can be created in the Project Browser, using the **Namespace** icon from the WSDL pages of the Toolbox. You can use this package structure as a template for developing your WSDL.

Access [View | Diagram Toolbox: More tools | WSDL](#)

Use to

- Create a new WSDL Package structure that can be used as a starting point for developing your WSDL model

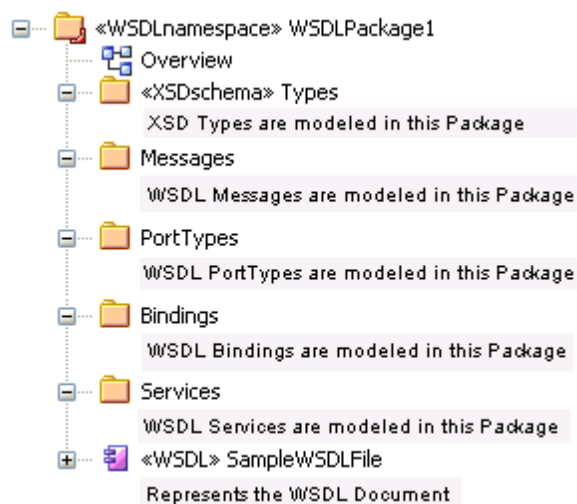
How to

Step	Action	See also

1	Create a package in the Project Browser.	
2	Click on the New Diagram icon in the Project Browser.	Project Browser Toolbar ^[583]
3	In the Select From field, select UML Structural , and in the Diagram Types field select Class .	
4	Click on the More tools option in the Toolbox and select WSDL from the list of Toolboxes.	Diagram Toolbox ^[698]
5	Click on the Namespace icon from the Toolbox and drag it into the created Class diagram.	WSDL Group ^[719]
6	Specify the WSDL Package Name and Namespace and click on the OK button to create the sample WSDL Package structure. <i>Alternatively:</i> A WSDL Package structure can also be created from an UML Interface using the WSDL Model Transformation.	Namespace Icon ^[2069] WSDL Model Transformation ^[1705]

Example WSDL Package Structure:

The *WSDLnamespace* Package acts as a container for the WSDL Structure in Enterprise Architect. The various WSDL constructs like *Messages*, *PortTypes*, *Bindings* and *Services* should be modeled in the **Messages**, **PortTypes**, **Bindings** and **Services** Packages respectively.



Notes

- A *WSDLnamespace* package can contain one or more *WSDL* stereotyped Components

- A *WSDL* stereotyped Component can be generated as a WSDL file

Learn more

- [WSDL Toolbox](#)^[719]
- [Model WSDL](#)^[2069]
- [Generate WSDL 1.1](#)^[2085]
- [Import WSDL 1.1](#)^[2086]

Learning Center topics

- (Alt+F1) | [SOA and XML Engineering | WSDL | Getting Started](#)

13.2.2 Model WSDL

The **Namespace** icon in the WSDL Toolbox creates a *WSDLnamespace* stereotyped Package which represents the top-level container for the WSDL constructs in Enterprise Architect.

WSDL Properties: Double-click on the *WSDLnamespace* stereotyped package in the diagram or the Project Browser

Reference

Field/Button	Usage	See also
WSDL Package Name	Specify the name for the WSDL package.	
Target Namespace	Specify the namespace for this WSDL package (<i>Optional</i>).	
OK	Save the values entered and close the dialog.	
Cancel	Discard the values entered and close the dialog.	
Help	Display this Help topic.	
UML	Open the UML Properties dialog. This button is available when editing the Schema package.	Properties ^[843]

Learn more

The *WSDLnamespace* stereotyped Package acts as a container for the following WSDL constructs:

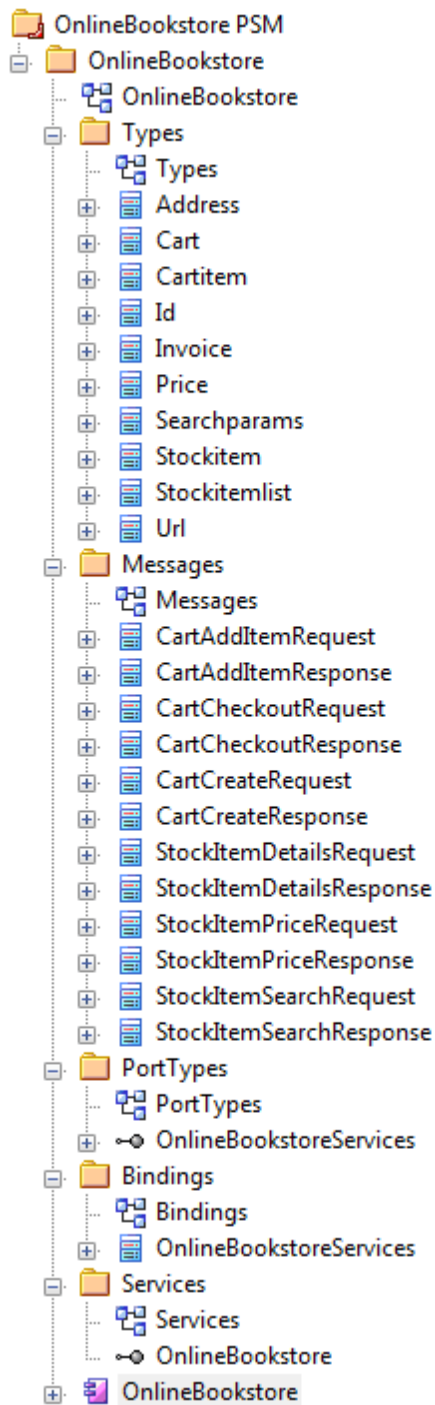
- [WSDL Namespace](#) ^[2070]
- [WSDL Message](#) ^[2073]
- [WSDL](#) ^[2073] [Message Part](#) ^[2074]
- [WSDL](#) ^[2073] [Port Type](#) ^[2075]
- [WSDL](#) ^[2073] [Port Type Operation](#) ^[2076]
- [WSDL](#) ^[2073] [Binding](#) ^[2078]
- [WSDL](#) ^[2073] [Binding Operation](#) ^[2080]
- [WSDL](#) ^[2073] [Service](#) ^[2082]
- [WSDL Document](#) ^[2083]

13.2.2.1 WSDL Namespace

The WSDL namespace in Enterprise Architect represents the top-level container for the WSDL elements, including WSDL documents. Conceptually it maps to the *targetNamespace* in a WSDL definition element. A given WSDL namespace can reuse its schema Types, Messages, Port Types, Bindings and Service across multiple physical WSDL documents.

The figure below shows an example WSDL namespace (*OnlineBookstore PSM*, which has a red margin to the bottom right corner), including a single WSDL document:

Example



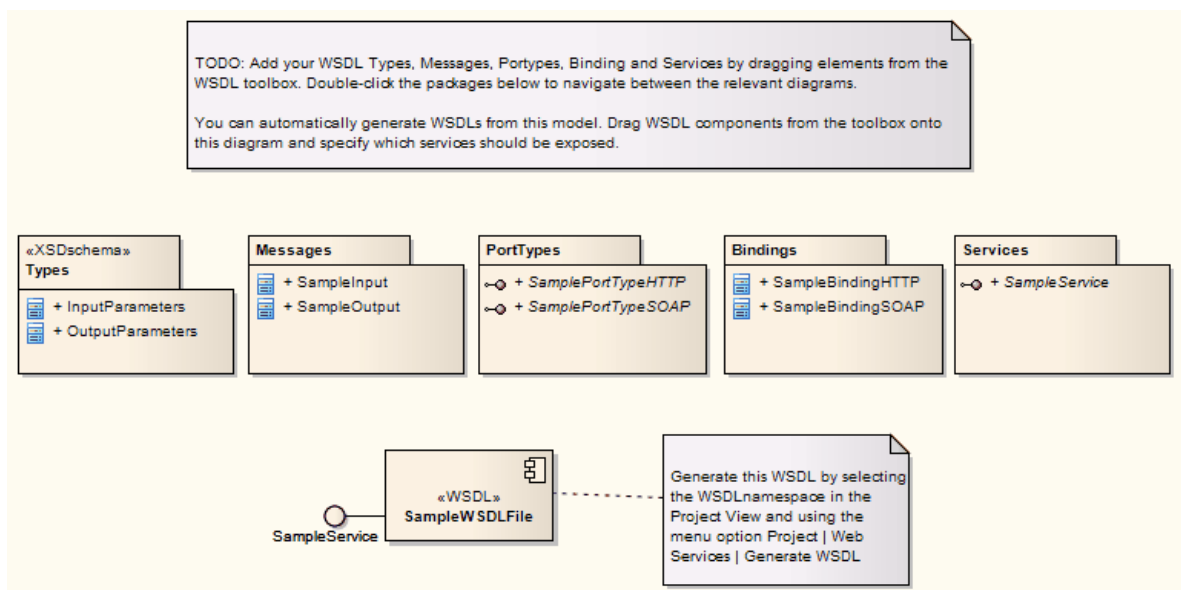
How to

To create a new WSDL namespace in your model

Step	Action	See also
1	Open or create the appropriate diagram.	

Step	Action	See also
2	Select the More tools WSDL menu option from the Toolbox.	
3	<p>Drag the Namespace element from the Toolbox onto the diagram.</p> <p>The WSDL Namespace Properties dialog displays:</p>	
4	Type in a WSDL Package Name and Target Namespace name. You can edit these values later.	
5	<p>Click on the OK button to create a package stereotyped as <i>WSDLnamespace</i>. This contains the following sub-packages and an Overview diagram to navigate between the sub-packages:</p> <ul style="list-style-type: none"> • Types: Contains the XSD types used by the WSDL Message elements; this package is modeled as an XML Schema, and you drag <i>XSDelement</i>, <i>XSDsimpleType</i> and <i>XSDcomplexType</i> elements onto the Types diagram from the XML Schema page of the Toolbox • Messages: Contains the WSDL Messages, modeled as UML Classes marked with the stereotype <i>WSDLmessage</i> • PortTypes: Contains the WSDL Port Types, modeled as UML interfaces marked with the stereotype <i>WSDLportType</i> • Bindings: Contains the WSDL Bindings, modeled as UML Classes that realize the <i>PortTypes</i> • Services: Contains the WSDL Services, modeled as UML interfaces with associations to each exposed <i>Binding</i> 	
6	<p>Use the Overview diagram to navigate between the subpackages, by double-clicking the relevant packages.</p> <p>You can edit the sample WSDL elements created in the previous step, or drag new items from the WSDL pages of the Toolbox onto the relevant diagrams.</p>	

Example



You can edit the WSDL-specific properties of the namespace later by double-clicking the package in the Project Browser. Alternatively, on the WSDL Namespace Properties dialog, click on the **UML** button to invoke the standard Properties dialog for a package. (This button does not display on the initial WSDL Namespace Properties dialog for a new Namespace element.)

Learn more

- [Model XSD](#) ^[2027]
- [UML Profile for XSD](#) ^[2055]
- [XML Schema Group](#) ^[720]

13.2.2.2 WSDL Message

The **Message** icon in the WSDL Toolbox creates a *WSDLmessage* stereotyped Class, representing the WSDL Message. The *WSDLmessage* stereotyped Class acts as a container for one or more WSDL Message Parts. WSDL Message should be defined under the *Messages* package in the WSDL Package structure.

WSDL Properties: Double-click on the *WSDLmessage* stereotyped Class in the diagram or the Project Browser

Reference

Field/Button	Usage	See also
Name	Specify the name for the WSDL Message.	
Documentation	Specify the notes for this element (<i>Optional</i>).	
OK	Save the values entered and close the dialog.	

Cancel	Discard the values entered and close the dialog.	
Help	Display this Help topic.	
UML	Open the UML Properties dialog. This button is available when editing the WSDL Message.	Properties ^[843]

Notes

- WSDL Messages can only be created under the *Messages* package in the WSDL Package structure
- The name of the WSDL Message should be unique among all the WSDL Messages within the WSDL

Learn more

- [Message Part](#) ^[2074]
- [WSDL Package Structure](#) ^[2070]

13.2.2.1 WSDL Message Part

The **Message Part** icon from the WSDL Toolbox creates an UML Attribute when dropped on a *WSDLmessage* stereotyped Class. In Enterprise Architect, the WSDL Message Part must be associated with an XSD Type.

WSDL Properties: Double-click on the UML Attribute in the *WSDLmessage* stereotyped Class in the diagram or the Project Browser

Reference

Field/Button	Usage	See also
Name	Specify the name for the WSDL Message Part.	
Type	Specify : <ul style="list-style-type: none"> • XSD <i>built-in dataType</i> using the drop-down list or • XSD <i>Element or Complex Type or SimpleType Class</i> <p>The XSD Types can be selected from :</p> <ul style="list-style-type: none"> • Types package under the WSDL Package Structure or • Any package in the Project Browser 	XML Schema ^[2027] WSDL Package Structure ^[2070]

OK	Save the values entered and close the dialog.	
Cancel	Discard the values entered and close the dialog.	
Help	Display this Help topic.	
UML	Open the UML Properties dialog. This button is available when editing the WSDL Message Part.	Properties ⁸⁸⁷

Notes

- *WSDLmessage* stereotyped Classes can contain WSDL Message Parts only

Learn more

- [WSDL Message](#) ²⁰⁷³

Learning Center topics

- (Alt+F1) | [SOA and XML Engineering | WSDL | Parts Referring External XSD](#)

13.2.2.3 WSDL Port Type

The **Port Type** icon in the WSDL Toolbox creates a *WSDLportType* stereotyped Interface, representing the WSDL PortType.

The *WSDLportType* stereotyped Interface describes the operations exposed by the WSDL and acts as a container for one or more WSDL PortType Operations. WSDL PortType should be defined under the *PortTypes* package in the WSDL Package structure.

WSDL Properties: Double-click on the *WSDLportType* stereotyped Interface in the diagram or the Project Browser

Reference

Field/Button	Usage	See also
Name	Specify the name for the WSDL PortType.	
Documentation	Specify the notes for this element (<i>Optional</i>).	
OK	Save the values entered and close the dialog.	

Cancel	Discard the values entered and close the dialog.	
Help	Display this Help topic.	
UML	Open the UML Properties dialog. This button is available when editing the WSDL PortType.	Properties ^[843]

Notes

- WSDL PortTypes can only be created under the *PortTypes* package in the WSDL Package structure
- The name of the WSDL PortType should be unique among all the WSDL PortTypes within the WSDL

Learn more

- [WSDL PortType Operation](#) ^[2076]
- [WSDL Package Structure](#) ^[2070]

13.2.2.3.1 WSDL Port Type Operation

The **PortType Operation** icon from the WSDL Toolbox creates an UML Operation when dropped on a *WSDLportType* stereotyped Interface. Enterprise Architect supports creating the following PortType Operations:

- *One-Way*
- *Request-Response*
- *Solicit-Response*
- *Notification*

WSDL Properties: Double-click on the UML Operation in the *WSDLportType* stereotyped Interface in the diagram or the Project Browser


Reference

Field/Button	Usage	See also
Name	Specify the name for the WSDL PortType Operation.	
Documentation	Specify the notes for this element (<i>Optional</i>).	
Operation Type	Select one of the supported PortType Operation types: <ul style="list-style-type: none"> • The Output section is disabled for <i>One-Way</i> Operation Type 	

	<ul style="list-style-type: none"> The Input section is disabled for <i>Notification</i> Operation Type 	
Input	Specify the details of the input WSDL Message:	
	Field	Usage
	Name	Specify the name of the input Message. Defaults to the Operation Type .
	Message	Select a <i>WSDLmessage</i> stereotyped (Optional). The list displays all the WSDL Message in the <i>Message</i> package.
	Documentation	Specify the notes for this input Message (Optional).
Output	Specify the details of the output WSDL Message:	
	Field	Usage
	Name	Specify the name of the output Message. Defaults to the Operation Type .
	Message	Select a <i>WSDLmessage</i> stereotyped (Optional). The list displays all the WSDL Message in the <i>Message</i> package.
	Documentation	Specify the notes for this output Message (Optional).
Faults	Specify the details of the WSDL Message that can act as Faults:	
	Button	Usage
	New	Add a new Fault.
	Delete	Delete an existing Fault in the list.

[Message](#) ^[2073]

[WSDL Package Structure](#) ^[2070]

OK	Save the values entered and close the dialog.	
Cancel	Discard the values entered and close the dialog.	
Help	Display this Help topic.	
UML	Open the UML Properties dialog. This button is available when editing the WSDL PortType Operation.	Properties 

Notes

- WSDL PortType Operations can only be contained by WSDL PortTypes
- The value specified in the **Name** field of the **Input**, **Output** and **Fault** section above must be unique among all the Inputs, Outputs and Faults of the WSDL PortType respectively

Learn more

- [WSDL Port Type](#) 

Learning Center topics

- (Alt+F1) | [SOA and XML Engineering | WSDL | Types Referring External XSD](#)

13.2.2.4 WSDL Binding

The **Binding** icon in the WSDL Toolbox creates a *WSDLbinding* stereotyped Class, representing the WSDL Binding.



Each *WSDLbinding* Class implements the operations specified by a particular *WSDLportType* Interface and describes the message format and protocol details for the operations and messages defined by this *WSDLportType*.

WSDL Bindings should be defined under the *Bindings* package in the WSDL Package structure.

WSDL Properties: Double-click on the *WSDLbinding* stereotyped Class in the diagram or the Project Browser.

Reference

Field/Button	Usage	See also
Name	Specify the name for the WSDL PortType.	

PortType	Select the WSDL PortType to be implemented by this WSDL Binding.	Port Type 
Protocol	Specify the protocol for the transmission of the selected WSDL PortType's Operations. The supported protocols are: <ul style="list-style-type: none"> • <i>SOAP</i> • <i>HTTP</i> 	
Transport	Defines the SOAP protocol. This field is disabled for <i>HTTP</i> protocol. Default: <i>http://schemas.xmlsoap.org/soap/http</i>	
Style	Specify the style of SOAP protocol. This field is disabled for <i>HTTP</i> protocol.	
Verb	Specify the HTTP verb. The supported verbs are: <ul style="list-style-type: none"> • <i>GET</i> • <i>POST</i> This field is disabled for <i>SOAP</i> protocol.	
Documentation	Specify the notes for this element (<i>Optional</i>).	
OK	Save the values entered and close the dialog.	
Cancel	Discard the values entered and close the dialog.	
Help	Display this Help topic.	
UML	Open the UML Properties dialog. This button is available when editing the WSDL Binding.	Properties 

Notes

- A WSDL Binding must implement a WSDL PortType; therefore, WSDL PortTypes should be defined before creating WSDL Bindings
- WSDL Bindings can only be created under the *Bindings* package in the WSDL Package structure

- The name of the WSDL Binding should be unique among all the WSDL Bindings within the WSDL

Learn more

- [WSDL Binding Operation](#)^[2080]
- [WSDL Package Structure](#)^[2070]

13.2.2.4.1 WSDL Binding Operation

When a *WSDLbinding* stereotyped Class is created using the WSDL Binding dialog, Enterprise Architect draws a Realization connector from the WSDL Binding to the WSDL Porttype and automatically populates the WSDL Binding with all the UML Operations in the WSDL PortType.

WSDL Properties: Double-click on the UML Operation in the *WSDLbinding* stereotyped Class in the diagram or the Project Browser

Reference

Field/Button	Usage	See also				
Operation Name	Specifies the name of the Operation. This value in this field cannot be edited.					
Action	Specify the SOAP Action header (<i>URI</i>) for this Operation. This field is disabled for <i>HTTP</i> protocol.					
Style	Specify the style of this Operation. This field is disabled for <i>HTTP</i> protocol.					
Location	Specify the relative URI of this Operation. This field is disabled for SOAP protocol.					
Documentation	Specify the notes for this element (<i>Optional</i>).					
Parameters	Specify the details for the Operation Parameters. Click on the Details button to specify the details for Input, Output and Faults:					
	<table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Use</td> <td>Specify the encoding that is to be used. This field is disabled for <i>HTTP</i> protocol.</td> </tr> </tbody> </table>	Field	Usage	Use	Specify the encoding that is to be used. This field is disabled for <i>HTTP</i> protocol.	
Field	Usage					
Use	Specify the encoding that is to be used. This field is disabled for <i>HTTP</i> protocol.					

	<table border="1"> <tr> <td>Encoding Style</td> <td>Specify the style (URIs) when Use is set to <i>encode</i>. This field is disabled for <i>HTTP</i> protocol.</td> </tr> <tr> <td>Namespace</td> <td>Specify the namespace (<i>Optional</i>). This field is disabled for <i>HTTP</i> protocol.</td> </tr> <tr> <td>Parts</td> <td>Specifies the Message Parts that appear within the Body portion (<i>Optional</i>). This field is disabled for <i>HTTP</i> protocol.</td> </tr> <tr> <td>Header</td> <td>Specify the contents of the SOAP/HTTP Header (<i>Optional</i>).</td> </tr> <tr> <td>Documentation</td> <td>Specify the notes for this input Message (<i>Optional</i>).</td> </tr> </table> <ul style="list-style-type: none"> • Details in the Input section is disabled for the <i>Notification</i> Operation Type • Details in the Output section is disabled for the <i>One-way</i> Operation Type • Details in the Fault section is disabled where there are no Fault Messages 	Encoding Style	Specify the style (URIs) when Use is set to <i>encode</i> . This field is disabled for <i>HTTP</i> protocol.	Namespace	Specify the namespace (<i>Optional</i>). This field is disabled for <i>HTTP</i> protocol.	Parts	Specifies the Message Parts that appear within the Body portion (<i>Optional</i>). This field is disabled for <i>HTTP</i> protocol.	Header	Specify the contents of the SOAP/HTTP Header (<i>Optional</i>).	Documentation	Specify the notes for this input Message (<i>Optional</i>).	
Encoding Style	Specify the style (URIs) when Use is set to <i>encode</i> . This field is disabled for <i>HTTP</i> protocol.											
Namespace	Specify the namespace (<i>Optional</i>). This field is disabled for <i>HTTP</i> protocol.											
Parts	Specifies the Message Parts that appear within the Body portion (<i>Optional</i>). This field is disabled for <i>HTTP</i> protocol.											
Header	Specify the contents of the SOAP/HTTP Header (<i>Optional</i>).											
Documentation	Specify the notes for this input Message (<i>Optional</i>).											
OK	Save the values entered and close the dialog.											
Cancel	Discard the values entered and close the dialog.											
Help	Display this Help topic.											
UML	Open the UML Properties dialog. This button is available when editing the WSDL Binding Operation.	Properties ☞										

Learn more

- [WSDL Binding](#)^[2078]

13.2.2.5 WSDL Service

The **Service** icon in the WSDL Toolbox creates a *WSDLservice* stereotyped Interface, representing the WSDL Service.

The *WSDLservice* stereotyped Interface describes the collection of ports that expose a particular Binding. WSDL Service should be defined under the *Services* package in the WSDL Package structure.

WSDL Properties: Double-click on the *WSDLservice* stereotyped Interface in the diagram or the Project Browser

Reference

Field/Button	Usage	See also														
Name	Specify the name for the WSDL Service.															
Documentation	Specify the notes for this element (<i>Optional</i>).															
Ports	Specify the Ports (or endpoints) for this WSDL Service.															
	<table border="1"> <thead> <tr> <th>Button</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Click on the New button to add a Port using the WSDL Port dialog.</td> <td></td> </tr> </tbody> </table>	Button	Usage	Click on the New button to add a Port using the WSDL Port dialog.												
	Button	Usage														
	Click on the New button to add a Port using the WSDL Port dialog.															
	<table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Port Name</td> <td>Specify the name of the Port.</td> </tr> <tr> <td>Binding</td> <td>Select a <i>WSDLbinding</i> stereotyped Class The list displays all the WSDL Bindings of the <i>Bindings</i> package.</td> </tr> <tr> <td>Location</td> <td>Specify the URI for the Port.</td> </tr> <tr> <td>Documentation</td> <td>Specify the notes for this input Message (<i>Optional</i>).</td> </tr> <tr> <td>OK</td> <td>Save the values entered and close the WSDL Port dialog.</td> </tr> <tr> <td>Cancel</td> <td>Discard the values entered and close the WSDL Port dialog.</td> </tr> </tbody> </table>	Field	Usage	Port Name	Specify the name of the Port.	Binding	Select a <i>WSDLbinding</i> stereotyped Class The list displays all the WSDL Bindings of the <i>Bindings</i> package.	Location	Specify the URI for the Port.	Documentation	Specify the notes for this input Message (<i>Optional</i>).	OK	Save the values entered and close the WSDL Port dialog.	Cancel	Discard the values entered and close the WSDL Port dialog.	Binding ^[2078]
	Field	Usage														
	Port Name	Specify the name of the Port.														
	Binding	Select a <i>WSDLbinding</i> stereotyped Class The list displays all the WSDL Bindings of the <i>Bindings</i> package.														
	Location	Specify the URI for the Port.														
	Documentation	Specify the notes for this input Message (<i>Optional</i>).														
OK	Save the values entered and close the WSDL Port dialog.															
Cancel	Discard the values entered and close the WSDL Port dialog.															

	<table border="1"> <tr> <td>Help</td> <td>Display this Help topic.</td> </tr> <tr> <td colspan="2">Press the Delete key to remove an entry from the list.</td> </tr> </table>	Help	Display this Help topic.	Press the Delete key to remove an entry from the list.		
Help	Display this Help topic.					
Press the Delete key to remove an entry from the list.						
OK	Save the values entered and close the dialog.					
Cancel	Discard the values entered and close the dialog.					
Help	Display this Help topic.					
UML	Open the UML Properties dialog. This button is available when editing the WSDL Service.	Properties ^[843]				

Notes

- WSDL Services can only be created under the *Service* package in the WSDL Package structure
- The name of the WSDL Service should be unique among all the WSDL Services within the WSDL

Learn more

- [WSDL Package Structure](#) ^[2070]

13.2.2.6 WSDL Document

The **WSDL** icon in the WSDL Toolbox creates a *WSDL* stereotyped Component, representing the WSDL document.

The *WSDL* stereotyped Component should be modeled as direct child element of the *WSDLnamespace* stereotyped Package. Multiple WSDL documents can be created for a single *WSDLnamespace* stereotyped Package, thus enabling the WSDL Services for that namespace to be reused and exposed as required across multiple WSDLs.

WSDL Properties: Double-click on the *WSDL* stereotyped Component in the diagram or the Project Browser

Reference

Field/Button	Usage	See also
Name	Specify the name for the WSDL document.	

File Name	Specify the file path where the WSDL 1.1 file is to be generated.																	
Documentation	Specify the notes for this element (<i>Optional</i>).																	
XMLNS	<p>List specifying the additional namespace and namespace-prefix pairs used in this WSDL document.</p> <table border="1"> <thead> <tr> <th>Button</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td></td> <td>Click on the New button to add an entry in the list. Double-click on an entry in the list to edit the values.</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Prefix</td> <td>Specify the abbreviated value that represents the N</td> </tr> <tr> <td>Namespace</td> <td>Specify the namespace.</td> </tr> <tr> <td>OK</td> <td>Save the values entered and close the Namespace</td> </tr> <tr> <td>Cancel</td> <td>Discard the values entered and close the Namesp</td> </tr> <tr> <td>Help</td> <td>Display this Help topic.</td> </tr> </tbody> </table> <p>Click on the Delete button to remove an entry from the list.</p>	Button	Usage		Click on the New button to add an entry in the list. Double-click on an entry in the list to edit the values.	Field	Usage	Prefix	Specify the abbreviated value that represents the N	Namespace	Specify the namespace.	OK	Save the values entered and close the Namespace	Cancel	Discard the values entered and close the Namesp	Help	Display this Help topic.	
Button	Usage																	
	Click on the New button to add an entry in the list. Double-click on an entry in the list to edit the values.																	
Field	Usage																	
Prefix	Specify the abbreviated value that represents the N																	
Namespace	Specify the namespace.																	
OK	Save the values entered and close the Namespace																	
Cancel	Discard the values entered and close the Namesp																	
Help	Display this Help topic.																	
Services	<p>Select the required WSDL Services.</p> <p>The list displays all the WSDL Services created in the <i>Services</i> package.</p>																	
OK	Save the values entered and close the dialog.																	
Cancel	Discard the values entered and close the dialog.																	
Help	Display this Help topic.																	

UML	Open the UML Properties dialog. This button is available when editing the WSDL document.	Properties [843]

13.2.3 Generate WSDL

The **Generate WSDL** feature forward engineers a UML model to a WSDL 1.1 file. The **Generate WSDL** feature acts either on a *WSDLnamespace* stereotyped Package or a *WSDL* stereotyped Component. It is used to generate any or all of the *WSDL* stereotyped Components owned by the target *WSDLnamespace* structure.

Access **Tools | Web Service | Generate WSDL**, or
Right-click WSDL stereotyped Component | Generate WSDL

Reference

Field/Button	Usage	See also										
Encoding	Specify the encoding scheme for the WSDL file. Default: Value of the Code Page field of the XML Specifications section in the Options dialog.	XML Specifications [562]										
Select Components To Generate	List all the <i>WSDL</i> stereotyped Components under the current <i>WSDLnamespace</i> Package. Highlight the required <i>WSDL</i> stereotyped Components for which the WSDL file is to be generated. <table border="1" data-bbox="480 1375 1236 1637"> <thead> <tr> <th>Button</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Select All</td> <td>Selects all the entries in this list.</td> </tr> <tr> <td>Select None</td> <td>Clear all selected entries in this list.</td> </tr> </tbody> </table> Double-click on an entry in the list to open the Component File Name dialog: <table border="1" data-bbox="480 1823 1236 1982"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Specifies the name of the <i>WSDL</i> stereotyped Comp</td> </tr> </tbody> </table>	Button	Usage	Select All	Selects all the entries in this list.	Select None	Clear all selected entries in this list.	Field	Usage	Name	Specifies the name of the <i>WSDL</i> stereotyped Comp	
Button	Usage											
Select All	Selects all the entries in this list.											
Select None	Clear all selected entries in this list.											
Field	Usage											
Name	Specifies the name of the <i>WSDL</i> stereotyped Comp											

Field/Button	Usage	See also				
	<table border="1"> <tr> <td>Prefix</td> <td>Select the required prefix for the WSDL namespace prefixes had been specified.</td> </tr> <tr> <td>File Name</td> <td>Specify the file path and name into which the WSDL generated.</td> </tr> </table>	Prefix	Select the required prefix for the WSDL namespace prefixes had been specified.	File Name	Specify the file path and name into which the WSDL generated.	
Prefix	Select the required prefix for the WSDL namespace prefixes had been specified.					
File Name	Specify the file path and name into which the WSDL generated.					
Generate	Generate the WSDL file for all the selected <i>WSDL</i> stereotyped Components.					
View WSDL	View the generated WSDL for a selected <i>WSDL</i> stereotyped Component.					
Close	Close this dialog.					
Help	Display this Help topic.					
Progress	Lists the progress of WSDL generation.					

Learning Center topics

- (Alt+F1) | [SOA and XML Engineering | WSDL | Generate WSDL](#)

13.2.4 Import WSDL

Enterprise Architect provides a **WSDL Import** facility to reverse engineer WSDL 1.1 files into UML Class models.

Access [Tools | Web Service | Import WSDL](#), or
[Project Browser Package context menu | Code Engineering | Import WSDL](#)

Reference

Field/Button	Usage
Filename	Specify the WSDL file to import.
Root Package	Specify the Package under which the WSDL file is to be imported.

Field/Button	Usage
Target Package	Specify the name of the Package that represents the imported WSDL file. Default: Name of the WSDL file.
Import	Initiate the WSDL Import.
Close	Close this dialog.
Help	Display this Help topic.
Progress	Lists the progress of the WSDL Import.

Notes

Enterprise Architect cannot import a WSDL file that references WSDL constructs existing outside this WSDL file	
Importable Example	http://www.w3.org/TR/wsdl.html#_wsdl
Non-importable Example	http://www.w3.org/TR/wsdl.html#_style Attempting to import the second WSDL file would result in the following error: Cannot Import Split Files.
Workaround	Combine the split WSDL files into a single file and then import it into Enterprise Architect.

Learning Center topics

- (Alt+F1) | [SOA and XML Engineering | WSDL | Import WSDL](#)

13.3 SoaML

Topics

Topic	Detail	See also
Abstract	<p>The following text is derived from Service oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and metamodel for Services (UPMS) (OMG document ad/2008-11-01); pp. 25-26:</p> <p><i>A service is an offer of value to another through a well-defined interface and available to a community (which may be the general public). A service results in work provided to one by another.</i></p> <p><i>Service Oriented Architecture (SOA) is a way of organizing and understanding (representations of) organizations, communities and systems to maximize agility, scale and interoperability. The SOA approach is simple - people, organizations and systems provide services to each other. These services allow us to get something done without doing it ourselves or even without knowing how to do it - enabling us to be more efficient and agile. Services also enable us to offer our capabilities to others in exchange for some value - thus establishing a community, process or marketplace. The SOA paradigm works equally well for integrating existing capabilities as for creating and integrating new capabilities.</i></p> <p><i>SOA ... is an architectural paradigm for defining how people, organizations and systems provide and use services to achieve results. SoaML ... provides a standard way to architect and model SOA solutions using the Unified Modeling Language (UML). The profile uses the built-in extension mechanisms of UML to define SOA concepts in terms of existing UML concepts.</i></p> <p><i>... the highest leverage of employing SOA comes from understanding a community, process or enterprise as a set of interrelated services and ... supporting that service oriented enterprise with service-enabled systems. SoaML enables business oriented and systems oriented services architectures to mutually and collaboratively support the enterprise mission. ... SoaML depends on Model Driven Architecture® (MDA®) to help map business and systems architectures, the design of the enterprise, to the technologies that support SOA, like web services and CORBA®.</i></p> <p>For further information on the concepts of SoaML, see the specification document on the OMG website SOaML document page.</p>	<p>SOaML document page (Online Resource)</p>
SoaML in Enterprise Architect	<p>Enterprise Architect enables you to model services architectures quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer. The SoaML facilities are provided in the form of:</p> <ul style="list-style-type: none"> • Two SoaML diagram types - <i>SoaML Component Diagram</i> and <i>SoaML Sequence Diagram</i> - accessed through the New Diagram dialog • SoaML pages in the Toolbox 	<p>New Diagram ^[72]</p> <p>Toolbox</p> <p>Shortcut Menu ^[70]</p> <p>Quick Linker ^[79]</p>

Topic	Detail	See also
	<ul style="list-style-type: none">• SoaML element and relationship entries in the Toolbox Shortcut menu and Quick Linker	
SoaML Toolbox Pages	You can access the SoaML pages of the Toolbox through the More tools SoaML menu option. There is a set of pages for each SoaML diagram type, although the last five pages in each set are the same.	

Topic	Detail	See also
	<p>SoaML Component Diagram</p> <ul style="list-style-type: none"> UML Component <ul style="list-style-type: none"> Package Packaging Component Component Class Interface Object Part Expose Interface UML Component Relationships <ul style="list-style-type: none"> Assembly Delegate Associate Realize Generalize Nesting SOA - Contracts <ul style="list-style-type: none"> Collaboration Service Contract Services Architecture Participant Architecture... Collaboration Use SOA - Services <ul style="list-style-type: none"> Agent... Entity ID... Participant... Service Interface... Specification Port Request Point Service Point Service Channel SOA - Service Data <ul style="list-style-type: none"> Attachment... Message Type SOA - Milestones <ul style="list-style-type: none"> Milestone SOA - Capabilities <ul style="list-style-type: none"> Capability... Expose 	<p>SoaML Sequence Diagram</p> <ul style="list-style-type: none"> UML Sequence <ul style="list-style-type: none"> Lifeline Fragment Endpoint Diagram Gate State/Continuation UML Sequence <ul style="list-style-type: none"> Message Self-Message Call Recursion SOA - Contracts SOA - Services SOA - Service Data SOA - Milestones SOA - Capabilities Common

Topic	Detail	See also
<p>Example SoaML Diagram</p>	<p>The diagram illustrates a SoaML service architecture. At the top level, a dashed oval represents the «servicesArchitecture» DealerNetworkArchitecture. Inside this oval, a central «collaboration» ps :PurchasingService is shown. Two roles, buyer and seller, are connected to this collaboration. The buyer role is associated with a participant «participant» dealer, and the seller role is associated with a participant «participant» mfg. Below the oval, two detailed participant boxes are shown. The left box is for «participant» Dealer, which includes a «servicePoint» Buyer. The right box is for «participant» Manufacturer, which includes a «servicePoint» ServicePoint. Dashed arrows labeled type connect the top-level participants to their detailed counterparts. A «Comment» box in the center of the oval states: "The purpose of the dealer network architecture is to establish the financially related services between dealers and manufacturers such that any dealer can do business with any manufacturer."</p>	
<p>Disable SoaML</p>	<p>If you prefer not to use SoaML in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).</p>	<p>Manage MDG Technologies [1285]</p>


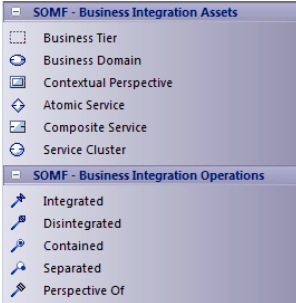
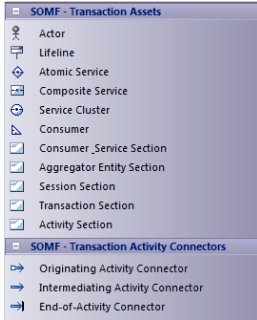
Notes

- Service Oriented Architecture Modeling Language (SoaML) is supported in the Corporate, Systems Engineering, Business and Software Engineering and Ultimate editions of Enterprise Architect

13.4 SOMF

Topics

Topic	Detail	See also
Abstract	<p>The following text is derived from the extensive Wikipedia entry on Service Oriented Modeling:</p> <p><i>The Service-Oriented Modeling Framework (SOMF) has been proposed by author Michael Bell as a holistic and anthropomorphic modeling language for software development that employs disciplines and a universal language to provide tactical and strategic solutions to enterprise problems. The term "holistic language" pertains to a modeling language that can be employed to design any application, business and technological environment, either local or distributed. This universality may include design of application-level and enterprise-level solutions, including SOA landscapes or Cloud Computing environments. The term "anthropomorphic", on the other hand, affiliates the SOMF language with intuitiveness of implementation and simplicity of usage.</i></p> <p><i>The service-oriented modeling framework (SOMF) is a service-oriented development life cycle methodology. It offers a number of modeling practices and disciplines that contribute to a successful service-oriented life cycle management and modeling. ... It illustrates the major elements that identify the "what to do" aspects of a service development scheme. These are the modeling pillars that will enable practitioners to craft an effective project plan and to identify the milestones of a service-oriented initiative- either a small or large-scale business or a technological venture.</i></p> <p>For further information on the concepts of SOMF, see the Wikipedia Service Oriented Modeling pages.</p>	<p>Michael Bell (Online Resource)</p> <p>Service Oriented Modeling (Online Resource)</p>
SOMF in Enterprise Architect	<p>In Enterprise Architect, SOMF is implemented as a profile within an MDG Technology that is integrated with the Enterprise Architect installer. For further information on this MDG Technology, see the <i>Service-Oriented Modeling Framework™ in Enterprise Architect</i> page on the Sparx Systems website.</p> <p>The SOMF facilities are provided in the form of:</p> <ul style="list-style-type: none"> Eight SOMF diagram types accessed through the New Diagram dialog; these are SOMF: <ul style="list-style-type: none"> <i>Analysis</i> <i>Business Integration</i> <i>Logical Design Relationship</i> <i>Logical Design Composition</i> <i>Transaction</i> <i>Conceptual Architecture</i> <i>Asset Utilization</i> <i>Transaction Directory</i> SOMF pages in the Toolbox 	<p>Sparx Systems (online resource)</p> <p>New Diagram <small>[724]</small></p> <p>Toolbox Shortcut Menu <small>[704]</small></p> <p>Quick Linker <small>[791]</small></p>

Topic	Detail	See also
	<ul style="list-style-type: none"> SOMF element and relationship entries in the Toolbox Shortcut menu and Quick Linker 	
SOMF Toolbox Pages	<p>You can access the SOMF pages of the Toolbox through the More tools SOMF menu option. There is a set of pages for each SOMF diagram type</p> <div style="display: flex; justify-content: space-around;"> <div data-bbox="395 577 639 633" style="text-align: center;"> <p>SOMF Conceptual Architecture Diagram</p>  </div> <div data-bbox="703 577 927 633" style="text-align: center;"> <p>SOMF Business Integration Diagram</p>  </div> <div data-bbox="1018 577 1230 633" style="text-align: center;"> <p>SOMF Transaction Diagram</p>  </div> </div>	
Disable SOMF	<p>If you prefer not to use SOMF in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).</p>	MDG Technologies ¹²⁸⁵

13.5 Generate MOF

Enterprise Architect offers support for exporting packages to XML under the *Meta-Object Facility (MOF)* 1.3 and 1.4 standards.

MOF models are created by assigning the stereotype *metamodel* to the package.

MOF models can be exported to MOF 1.3 or MOF 1.4 XML file specifications.

Topics

Topic	Detail	See also
Background Knowledge	<p>MOF is an Object Management Group (OMG) standard that originated in the UML, when the OMG required a Meta-Modeling architecture to define the UML. MOF is designed as a four-layered architecture, as illustrated in the following diagram.</p> <pre> classDiagram class Classifier { +type } class GeneralizableElement { -isAbstract: Boolean -isLeaf: Boolean -isRoot: Boolean -visibility: visibilityKind -allSupertypes() -findElementsByTypeExtended() -lookupElementExtended() } class NameSpace { -findElementsB() -lookupElement() -namesValid() -resolveQualifiedName() } class ModelElement { -annotation: String -name: String -qualifiedName: String -findRequiredElements() -isFrozen() -isRequiredBecause() -isVisible() } class TypedElement { +typedElement } Classifier < -- GeneralizableElement GeneralizableElement < -- NameSpace GeneralizableElement < -- TypedElement NameSpace < -- ModelElement TypedElement < -- ModelElement Classifier "1" --> "0..*" GeneralizableElement : +type GeneralizableElement "0..*" --> "0..*" ModelElement : +dependent {ordered} NameSpace "0..*" --> "0..*" ModelElement : +provider TypedElement "0..*" --> "0..*" ModelElement : +typedElement </pre> <p>Because of the similarities between the MOF-model and UML structure models, MOF meta-models are usually modeled as UML</p>	<p>Class Diagrams ^[1014]</p> <p>Metamodel Group ^[715]</p>

Topic	Detail	See also
	<p>Class diagrams. You can also use the Metamodel page of the Toolbox to create MOF model elements and connectors. A supporting standard of MOF is XML, which defines an XML-based exchange format.</p> <p>MOF is a closed, strict meta-modeling architecture; every model element on every layer is strictly an instance of a model element of the layer above. MOF only provides a means to define the structure or abstract syntax of a languages or of data.</p> <p>Simplified, MOF uses the notion of Classes, as known from object orientation, to define concepts (model elements) on a meta-layer. These Classes (concepts) can then be instantiated through objects (instances) of the model layer below. Because an element on the M2 layer is an object (instance of an M3 model element) as well as a Class (an M2 layer concept) the notion of a <i>clabject</i> is used. <i>Clabject</i> is a merge of the words Class and Object.</p> <p>Another related standard is OCL, which describes a formal language that can be used to define model constraints by means of predicate logic.</p>	

Learn more

- [Getting Started](#)  ²⁰⁹⁵
- [Export MOF to XML](#)  ²⁰⁹⁷

Learning Center topics

- (Alt+F1) | [SOA and XML Engineering | MOF](#)

13.5.1 Getting Started

MOF diagrams are **Class** diagrams that are contained in packages with a metamodel stereotype.

How to

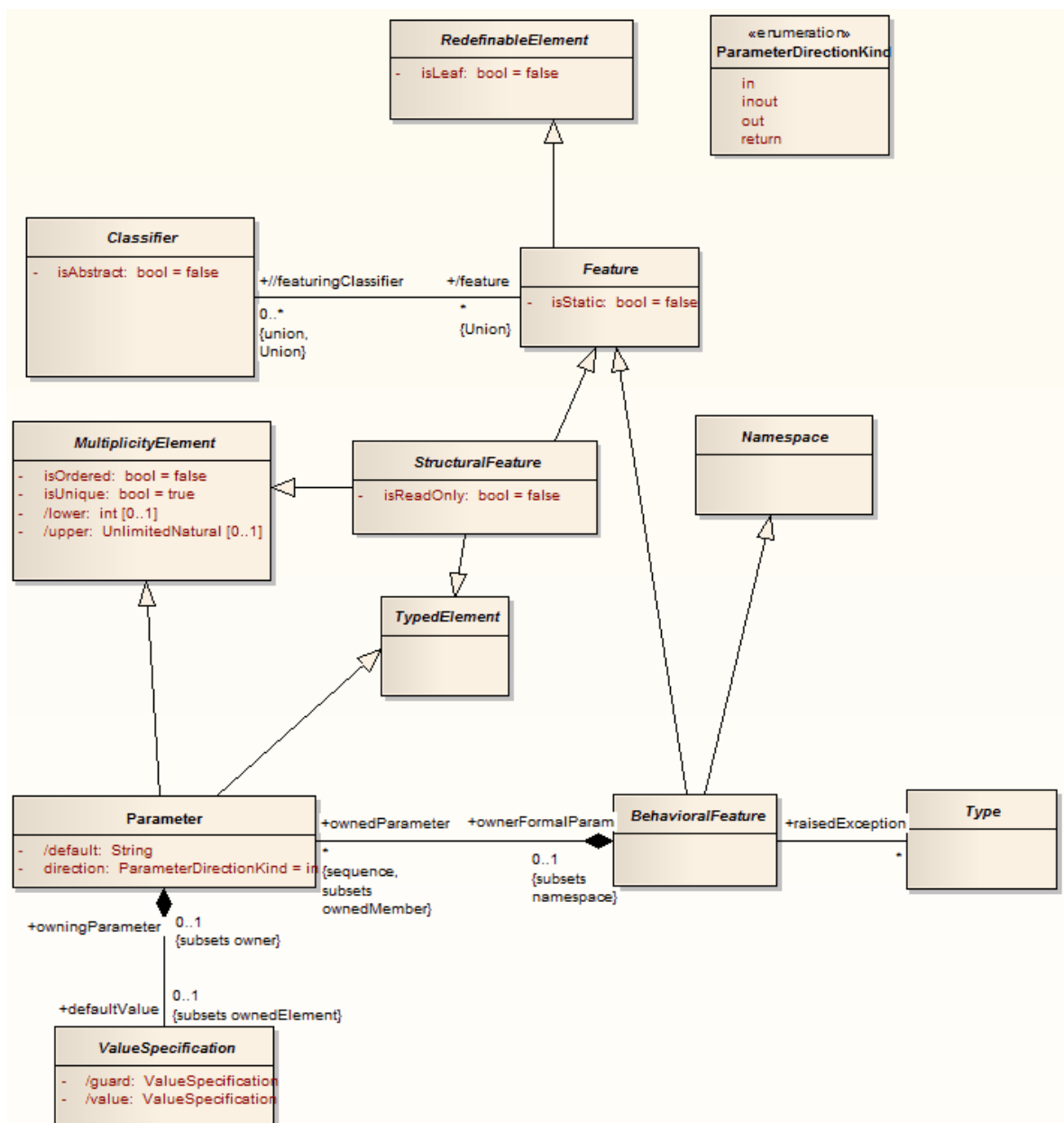
To create a MOF diagram

Step	Action	See also
1	Create a package to contain your MOF elements.	
2	Double-click on the package name to display the Properties dialog.	

Step	Action	See also
3	In the Stereotype field type the value metamodel .	
4	Click on the OK button.	
5	Right-click on the package in the Project Browser and select the Add Add Diagram context menu option. Create a Class diagram (the default diagram).	
6	Give your MOF Class diagram an appropriate name.	
7	In the Toolbox, select the More tools Metamodel menu option and add the required Metamodel elements to the diagram.	Metamodel Group ¹⁷¹⁵

Example

The following is an example of a MOF diagram. A MOF diagram can typically contain Package, Class, Enumeration and Primitive elements, and Generalization, Association, Compose and Aggregate relationships.



Learn more

- [Class Diagram](#) ^[1014]

Learning Center topics

- (Alt+F1) | [SOA and XML Engineering | MOF | Create a MOF Model](#)

13.5.2 Export MOF to XMI

How to

To export a MOF diagram to XMI, specifying the MOF 1.3 or MOF 1.4 standard

Step	Action	See also
1	Right-click on the package in the Project Browser and select the Import/Export Export Package to XMI file menu option. The Export Package to XMI dialog displays.	
2	In the Filename field, type a name for the XMI file.	
3	De-select the Enable full EA Roundtrip checkbox.	
4	In the XMI Type field, click on the drop-down arrow and select MOF 1.3 or MOF 1.4 .	
5	Click on the Export button and wait until the Progress bar reads 100% .	
6	Once your file has been created, you can view it by clicking on the View XMI button.	Metamodel Group 7151

Notes

- MOF diagrams exported to XMI can be imported using the regular import XMI features of Enterprise Architect

Learn more

- [Import from XMI](#)⁴¹⁷

Learning Center topics

- (Alt+F1) | **SOA and XML Engineering | MOF | Export MOF to XMI**

Part



14 Model Simulation

Model Simulation brings your **behavioral models to life** with instant, real-time **behavioral model execution**. Coupled with tools to manage triggers, events, guards, effects, breakpoints and simulation variables - plus the ability **visually track execution at run-time**, the Simulator is a powerful means of "watching the wheels turn" and verifying the correctness of your behavioral models.

With Simulation you can explore and **test the dynamic behavior of models**. In the Corporate and above editions, this also includes the ability to **use Javascript as a run-time execution language** for evaluating guards, effects and other script-able pieces of behavior.

- Extensive support for **triggers**, trigger sets, nested states, **concurrency**, dynamic **effects** and other **advanced simulation capabilities**, provides a remarkable environment in which to build **interactive** and **working models** which help explore, test and visually trace complex business, software and system **behavior**.
- With **Javascript** enabled, it is also possible to create embedded COM objects which will do the work of evaluating guards and executing effects - allowing the simulation to be tied into a much larger set of dependent processes. For example, a COM object evaluating a guard condition on a state transition might query a locally running process, read and use a set of test data - or even connect to a SOA web service to obtain some current information.
- As Enterprise Architect uses a **dynamic, script driven Simulation** mechanism there is no need to generate code or compile your model before running a simulation. It is even possible to update simulation variables in real time using the Simulation console window. This is useful for testing alternate branches and conditions "on the fly", either at a set Simulation break point, or when the Simulation reaches a point of stability (i.e. when the Simulation is 'blocked').
- The Professional version of Enterprise Architect includes the ability to manually walk through simulations - although no Javascript will execute - so all choices are manual decisions. This is useful for testing the flow of a behavioral model and highlighting possible choices and processing paths.

Simulation Overview

Topic	Link
Overview of the Model Simulator	How It Works ^[2102] How it Looks ^[2104]
Use of the Simulation window and related windows, and running a simulation	Simulation Windows ^[2106] Run Model Simulation ^[2112]
How to set up a simulation and activate a simulation script	Set Up Simulation Script ^[2109] Activate Simulation Script ^[2111]
How to set up and use simulation breakpoints	Simulation Breakpoints ^[2114]
Perform dynamic simulation with Javascript	Dynamic Simulation with Javascript ^[2116]

Topic	Link
The use of Guards and Effects in simulations	Guards and Effects ^[2119]
The use of Triggers in simulations	Triggers ^[2122] Simulation Events Window ^[2125] Waiting Triggers ^[2129] Re-Signal Triggers ^[2130] Trigger Parameters ^[2131] Trigger Sets and Auto-Firing ^[2133] Using Trigger Sets to Simulate an Event Sequence ^[2136]
Simulating multi-threaded processes	Multi-threading - Forks and Joins ^[2137] Multi-threading - Concurrent State Regions ^[2138]
Performing BPMN Simulations	BPMN Simulation ^[2139]

Notes

- Currently Javascript is enabled for State machines and Activity graphs - but not for Interaction diagrams. A future build will include Javascript enabled Interaction diagrams

Learning Center topics

- (Alt+F1) | **Execution Analysis | Simulation**

14.1 How It Works

The **Model Simulator** enables you to simulate the **execution of conceptual model designs** containing behavior.

When you start a simulation, the current model package is analyzed and a dynamic simulation process spawned to execute the model. As the simulator analyzes and works with UML constructs directly, there is no requirement to generate intermediary code or compile simulation 'executables'. This results in a very **rapid and dynamic simulation environment** in which **changes can be made and tested very rapidly**.

To **get up and running with simulation** the only steps required are:

- **Build a behavioral diagram** (state or activity for manual or dynamic execution, sequence for manual interaction only)
- Optional: load the Simulation Workspace layout ... a fast way of bringing up all the frequently used Simulation windows
- **Press the simulator "Play" button!**

The **Professional edition** provides a quick and simple way to verify your design's behavior for logical correctness by **manually** stepping through a diagram.

In the **Corporate edition** and above it is possible to:

- **dynamically executing** your behavioral models
- **assess guards and effects** written in standard Javascript
- **define and fire triggers** into running simulations
- define and **use sets of triggers** to simulate different event sequences
- **auto-fire trigger** sets to simulate complex event histories without user intervention
- **update simulation variables** "on the fly" to change how simulations proceed
- **create and call COM** objects during Simulation to extend the Simulation's reach and input/output possibilities
- inspect Simulation variables at run time
- **set a script** 'prologue' for defining variables, constants and functions prior to execution
- use multiple Analyzer Scripts with differing 'prologues' for running the Simulation under widely differing conditions
- in the Business and Software Engineering and Ultimate Editions it is also possible to simulate **BPMN models**

Platforms and Editions Available

Topic	Detail	See also
Platforms Supported	The Model Simulator supports the following simulation platforms: <ul style="list-style-type: none"> • UML Basic • BPMN 	Set Up Simulation <small>[2109]</small>

Topic	Detail	See also
	<p>The Model Simulator currently supports the execution of UML Activity, Interaction, State Machine models and BPMN Business Processes.</p>	
Edition Support	<p>Model Simulation is available at different levels across the range of editions of Enterprise Architect:</p> <ul style="list-style-type: none">• Desktop - Simulation not available• Professional - Manual Simulation only• Corporate and above - Adds dynamic Javascript evaluation• Business & Software Engineering and Ultimate - Adds BPMN simulation.	

Learn more

- [Set Up Simulation Script](#)^[2109]
- [Run Model Simulation](#)^[2112]
- [Diagram Context Menu](#)^[687]

14.2 How it Looks

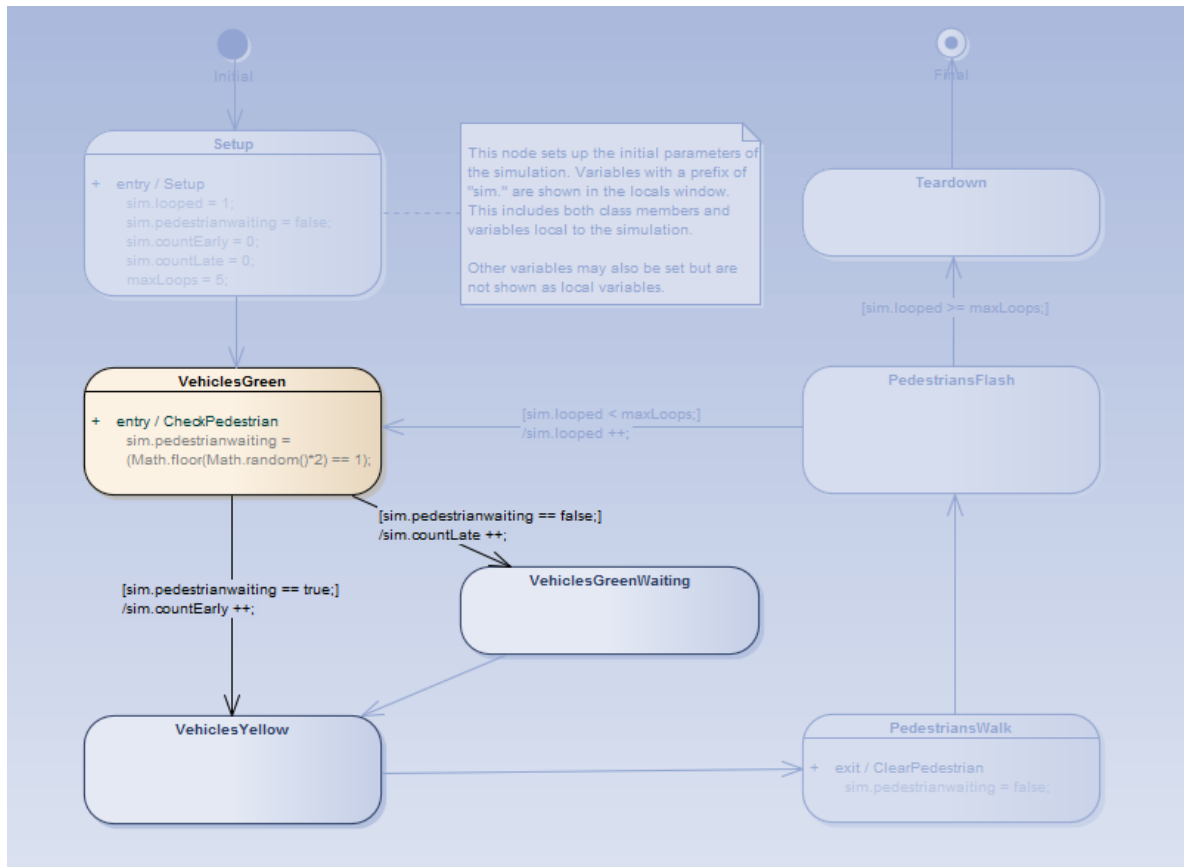
Enterprise Architect has a **special way of displaying model information during simulation**. This helps focus attention on the executing or active nodes.

During a Simulation, Enterprise Architect will **dynamically track and highlight the active nodes** within your model. If a node in another diagram is activated, that diagram will be automatically loaded and the current node highlighted. It is possible to modify the diagram while the simulation is running, however, the changes made are not recognized until the current simulation is ended and a new one begun.

Highlighting of the active node (s) during simulation:

- In the example below the currently active node ("VehiclesGreen") is highlighted in normal Enterprise Architect colors
- All possible **transitions out of the current node are rendered at full strength**
- The elements that are **possible targets** of the current active node's outgoing transitions are **rendered in a semi-faded** style so they are readable and clearly different to the other elements within the diagram
- All other elements are rendered in a "**fully faded**" style to show they are not targets of the next Simulation step.

As the Simulation progresses (especially if automatically run), this **highlighting helps focus the attention** on the current item and its visual context.

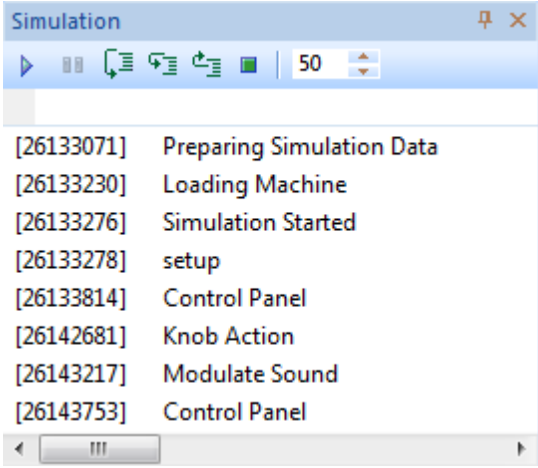


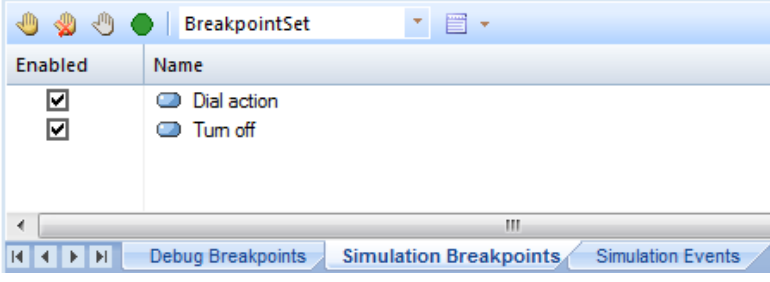
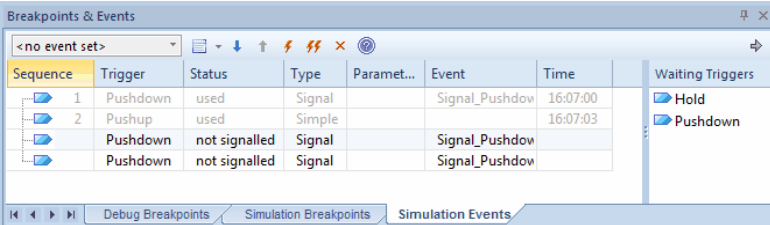

14.3 Simulation Windows

When executing a Simulation in Enterprise Architect it is possible to set **break-points**, fire **triggers**, examine **variables**, record a **trace** of execution, set simulation **speed**, view the **call stack** and **visually** trace the active nodes as the simulation proceeds.

When a Simulation runs, some aspects such as the output and console input are found in the Simulator window itself, while others such as the local variables and call-stack use the standard Execution Analyzer windows. The table below provides an overview of the main Windows used during Simulation.

Access [Analyzer | Simulation](#)

Window	Purpose	See also
Execution and Console	<p>The Simulation window provides the main interface for starting, stopping and stepping your Simulation. During execution it displays output relating to the currently executing step and other important information. See the "Run Model Simulation" topic for more information on the toolbar commands.</p> <p>Note the text entry box just underneath the toolbar. This is the Console input area - in here you can type simple javascript commands such as: <code>this.count = 4;</code> to dynamically change a Simulation variable named "count" to 4. In this way you can dynamically influence simulation at run-time.</p> 	Run Model Simulation ^[2112]
Simulation Breakpoints Window	<p>The Simulation window also makes use of the Simulation Breakpoints tab of the Breakpoints & Events window (Analyzer Breakpoints & Events). In here you set execution breakpoints on specific elements and messages in a Simulation. See the "Simulation Breakpoints" topic for more details.</p>	Simulation Breakpoints ^[2114]

Window	Purpose	See also
		
Simulation Events Window	<p>The Breakpoints and Events window also includes a tab for the Simulation Events window, which provides tools to manage and execute triggers. Triggers are used to control the execution of State machines transitions.</p> 	Simulation Events Triggers Waiting Triggers
Call Stack	<p>During the Simulation the Call Stack window displays information about the Threads and current execution context of the Simulation.</p> <p>The Simulator supports multi-threaded Simulations and will include a Thread entry for every active and paused thread of execution. For each thread, the Call Stack window will show the start or entry context (eg. a State Machine element) plus the current active element within that thread. If the current active element is the entry point of a composite activity or sub-machine state, the stack will also include the current active element within that sub-context (and all further nested, active composite, sub-states as well).</p> 	
Simulation Local Variable	<p>The Simulator uses the standard Local Variables window to show all current Simulation variables when the simulation is paused at a break point or single stepping. Note that it is possible to dynamically update these variables using the Simulator Console described above.</p>	

Window	Purpose	See also																																																																						
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #d3d3d3;">Variable</th> <th style="background-color: #d3d3d3;">Value</th> </tr> </thead> <tbody> <tr> <td>countEarly</td> <td>0</td> </tr> <tr> <td>countLate</td> <td>0</td> </tr> <tr> <td>looped</td> <td>1</td> </tr> <tr> <td>pedestrianwait</td> <td style="color: red;">True</td> </tr> </tbody> </table> <div style="border: 1px solid #ccc; width: 100%; height: 20px; margin-top: 10px;"></div>	Variable	Value	countEarly	0	countLate	0	looped	1	pedestrianwait	True																																																													
Variable	Value																																																																							
countEarly	0																																																																							
countLate	0																																																																							
looped	1																																																																							
pedestrianwait	True																																																																							
Recording	<p>During execution of your Simulation, a recording is kept of all activity and displayed in the Execution Analyzer recording window. This is similar to how the normal call recording works in the Visual Execution analyzer - although at this time, no ability to generate sequence diagrams from recordings is possible.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th>Sequence</th> <th>Instance</th> <th>Method</th> <th>Direction</th> <th>Method</th> <th>Instance</th> <th>Depth</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/> 00000001</td> <td></td> <td></td> <td>Call</td> <td>Pedestrian Crossing</td> <td></td> <td>1.1</td> </tr> <tr> <td><input checked="" type="checkbox"/> 00000002</td> <td></td> <td></td> <td>Call</td> <td>Pedestrian Crossing</td> <td></td> <td>1.1</td> </tr> <tr> <td><input checked="" type="checkbox"/> 00000003</td> <td></td> <td>Pedest...</td> <td>Call</td> <td>Setup</td> <td></td> <td>2.1</td> </tr> <tr> <td><input checked="" type="checkbox"/> 00000004</td> <td></td> <td>Pedest...</td> <td>Call</td> <td>VehiclesGreen</td> <td></td> <td>2.1</td> </tr> <tr> <td><input checked="" type="checkbox"/> 00000005</td> <td></td> <td>Pedest...</td> <td>Call</td> <td>VehiclesYellow</td> <td></td> <td>2.1</td> </tr> <tr> <td><input checked="" type="checkbox"/> 00000006</td> <td></td> <td>Pedest...</td> <td>Call</td> <td>PedestriansWalk</td> <td></td> <td>2.1</td> </tr> <tr> <td><input checked="" type="checkbox"/> 00000007</td> <td></td> <td>Pedest...</td> <td>Call</td> <td>PedestriansFlash</td> <td></td> <td>2.1</td> </tr> <tr> <td><input checked="" type="checkbox"/> 00000008</td> <td></td> <td>Pedest...</td> <td>Call</td> <td>VehiclesGreen</td> <td></td> <td>2.1</td> </tr> <tr style="background-color: #007bff; color: white;"> <td><input checked="" type="checkbox"/> 00000009</td> <td></td> <td>Pedest...</td> <td>Call</td> <td>VehiclesGreenWai...</td> <td></td> <td>2.1</td> </tr> </tbody> </table>	Sequence	Instance	Method	Direction	Method	Instance	Depth	<input checked="" type="checkbox"/> 00000001			Call	Pedestrian Crossing		1.1	<input checked="" type="checkbox"/> 00000002			Call	Pedestrian Crossing		1.1	<input checked="" type="checkbox"/> 00000003		Pedest...	Call	Setup		2.1	<input checked="" type="checkbox"/> 00000004		Pedest...	Call	VehiclesGreen		2.1	<input checked="" type="checkbox"/> 00000005		Pedest...	Call	VehiclesYellow		2.1	<input checked="" type="checkbox"/> 00000006		Pedest...	Call	PedestriansWalk		2.1	<input checked="" type="checkbox"/> 00000007		Pedest...	Call	PedestriansFlash		2.1	<input checked="" type="checkbox"/> 00000008		Pedest...	Call	VehiclesGreen		2.1	<input checked="" type="checkbox"/> 00000009		Pedest...	Call	VehiclesGreenWai...		2.1	
Sequence	Instance	Method	Direction	Method	Instance	Depth																																																																		
<input checked="" type="checkbox"/> 00000001			Call	Pedestrian Crossing		1.1																																																																		
<input checked="" type="checkbox"/> 00000002			Call	Pedestrian Crossing		1.1																																																																		
<input checked="" type="checkbox"/> 00000003		Pedest...	Call	Setup		2.1																																																																		
<input checked="" type="checkbox"/> 00000004		Pedest...	Call	VehiclesGreen		2.1																																																																		
<input checked="" type="checkbox"/> 00000005		Pedest...	Call	VehiclesYellow		2.1																																																																		
<input checked="" type="checkbox"/> 00000006		Pedest...	Call	PedestriansWalk		2.1																																																																		
<input checked="" type="checkbox"/> 00000007		Pedest...	Call	PedestriansFlash		2.1																																																																		
<input checked="" type="checkbox"/> 00000008		Pedest...	Call	VehiclesGreen		2.1																																																																		
<input checked="" type="checkbox"/> 00000009		Pedest...	Call	VehiclesGreenWai...		2.1																																																																		

14.4 Set Up Simulation Script

Simulation **scripts** are used to provide **fine control** over how a simulation starts. In general, **you do not need to set up a Simulation script unless you wish to:**

- An interpreted simulation requires variables to be **initialized** before the simulation commences
- The default behavior of interpreting the guards in Corporate Edition and above is undesired (ie. a **Manual execution is preferred**)
- **Multiple** ways of running the same diagram are required

For most diagrams it is possible to **start a simulation** simply by opening the diagram of interest and pressing the "Play" button on the Simulation control window.

To configure a simulation execution **script**, first select the **package** in either the **Project Browser, Package Browser, Diagram List or Model Search**. You can then use the Execution Analyzer window to add a new Script for the currently selected package. You will use the **Simulation page** of the Execution Analyzer dialog to configure the relevant properties.

Access **Analyzer | Execution Analyzer (Shift + F12) : New** (Toolbar icon)
Analyzer | Execution Analyzer (Shift + F12) : locate and double-click on required script >
Simulation

Execution Analyzer dialog Simulation page





Field/Option	Action	See also
Entry Point	<p>Click on the (...) button and select the required:</p> <ul style="list-style-type: none"> • Entry point for the simulation, and • Activity, Interaction or State Machine to simulate <p>Note that if you do not specify an entry point, the simulator attempts to simulate the entire package.</p>	
Platform	<p>For UML Activity, Interaction or State Machine simulation, click on the drop-down arrow and select UML Basic.</p> <p>For BPMN diagrams, click on the drop-down arrow and select BPMN.</p>	
Evaluate Guards and Effects using JavaScript	<p>If required, select the checkbox to execute the code for effect behavior in the simulation (instead of a manual simulation, where you select the next state to transition to and the point where a decision must be made) [note - this feature is only available in Corporate and above editions].</p> <p>The simulation executes javascript code in the following places:</p> <ul style="list-style-type: none"> • State entry/exit/do operations 	<p>State Machine Table Conventions ^[1047]</p> <p>Transition ^[1260]</p> <p>View the Local Variables ^[1877]</p>

Field/Option	Action	See also
	<ul style="list-style-type: none"> • Transition guard/effect • BPMN Activity Loop Conditions and Sequence Flow Condition Expressions. <p>With the exception of the guard, all should be one or more valid JavaScript statements, including the semi-colon.</p> <p>The guard must be a valid boolean expression, also terminated with a semi-colon.</p> <p>Variables that are initialized with a prefix of sim. or this. are available in the Locals window when a simulation breakpoint is hit <code>sim.count = 0</code>;</p>	
Input	When Javascript is enabled, you can enter script commands here that will execute prior to the simulation being run . This is useful setting up global variables and defining functions. By using multiple Analyzer Scripts with differing input scripts, you can run your simulation under a wide variety of conditions.	
OK	Click to save changes.	

Notes

- All simulation elements and relationships must reside within the package configured for simulation

Learn more

- [How Simulation Works](#) 
- [Activate Simulation Script](#) 
- [Run Model Simulation](#) 
- [Using Simulation Breakpoints](#) 

Learning Center topics

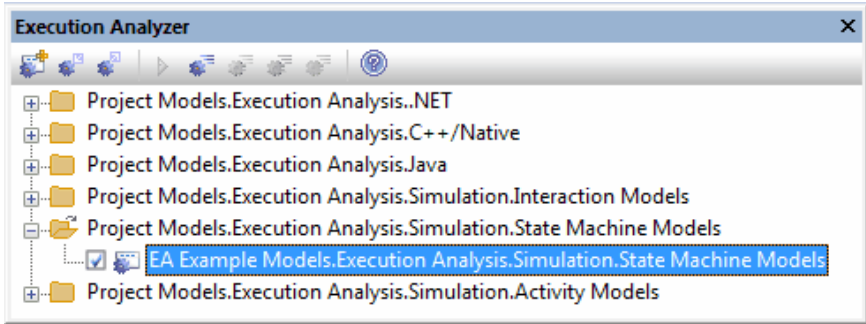
- **(Alt+F1) | Execution Analysis | Simulation | Configure a Simulation**

14.5 Activate Simulation Script

An Execution Script is configured for a model package defining the simulation parameters. The most common reason for activating an execution script is when multiple simulation scripts are configured against a package and you want to run a specific one.

Access [Project Browser Package Context Menu | Execution Analyzer | Analyzer Scripts...](#)

To activate a Simulation script for execution

Step	Action	See also
1	<p>In the Execution Analyzer window, select the required execution script. This makes it the current default for your open model, so that pressing the Simulation run button will automatically invoke this Simulation script.</p> 	
2	<p>Click on the checkbox to the left of the script to activate it.</p>	
3	<p>Select the Analyzer Simulation menu option to execute the simulation.</p>	Run Model Simulation ²¹¹²

Learn more

- [Set Up Simulation Script](#) ²¹⁰⁹

Learning Center topics

- (Alt+F1) | [Execution Analysis | Simulation | Run a Simulation](#)

14.6 Run Model Simulation

You **start a model simulation** in a number of ways

1. When the active diagram can be simulated the **"Run" button** on the main Simulation window will run the current diagram, either by running an existing script or defining a new temporary one.
2. When the active diagram can not be simulated the "Run" button on the main Simulation window will run the simulation for the active **Execution Analyzer script**.
3. By **right clicking on a Simulation script** in the Execution Analyzer window and selecting the "Start Simulation" option;
4. By **right clicking on a suitable diagram** and selecting one of the **Execute Simulation options** from the context menu.

Notes on **selecting a Simulation to run**






- You can set a Simulation script as the current default by setting its check box in the Execution Analyzer window.

Visual cues during execution. When the simulation is running, Enterprise Architect will **actively highlight each active node for each executed step**. In addition, all outgoing transitions, control flows & etc. will be highlighted, showing the possible paths forward. Elements at the end of possible paths forward will be de-emphasized to half-strength and any other remaining elements will be 90% "grayed out". This provides a very dynamic and easy to follow execution that continually refocuses attention on the execution 'context'.

Access Analyzer | Simulation

Running a Simulation

Topic	Detail	See also
Overview	<p>The simulation executes the model step-by-step, enabling you to validate the logic of your behavior model; the simulation automatically highlights the execution step in the model's diagram.</p> <p>In the Professional edition, if a branch is encountered in the execution, the simulator prompts you to choose the appropriate path to take in your execution.</p> <p>In the Corporate and above editions, when Javascript is enabled, the Simulation will automatically evaluate all guards, effects & etc. and dynamically execute the Simulation without user intervention. If the Simulation becomes blocked due to no possibly paths forward evaluating to true (or multiple paths evaluating to true) you can modify Simulation variables on the fly using the console input of the Simulation Execution window.</p>	<p>Activate Simulation Script [2111]</p>

Icon	Action	See also
	Start the simulator for the current simulatable diagram, or when the current diagram can't be simulated run simulation using the activated simulation script.	Activate Simulation Script ^[2111]
	Pause the simulation.	
	When the simulation is paused, step over , step in and step out to control the simulator's execution at the required step in the model simulation.	
	Stop the simulation.	
	Varies the execution rate of the simulation, between 0% and 100%; at: <ul style="list-style-type: none"> • 100%, the simulation executes at the fastest possible rate • 0% the simulator breaks execution at every statement 	

Notes

- The Simulation tool only becomes active when a valid simulation Execution Script is activated

Learn more

- [Set Up Simulation Script](#)^[2109]
- [Simulation Breakpoints](#)^[2114]

Learning Center topics

- (Alt+F1) | **Execution Analysis | Simulation | Run a Simulation**

14.7 Simulation Breakpoints

The **Simulation Breakpoints** tab of the **Breakpoints & Events window** enables you to **interrupt** and **inspect** the simulation process.



When **dynamically** executing a simulation, the process will proceed **automatically** (in the Corporate and above editions) - if you wish to stop execution at some point to examine variables, inspect call stacks or otherwise interact with the simulator, you can set a **breakpoint** on a model element in much the same way as you would with a line of source code. When the simulator reaches the breakpoint execution is **halted** and control returned to Enterprise Architect.



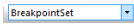

Access [Analyzer | Breakpoints & Events > Simulation Breakpoints](#)

Breakpoints

Topic	Detail	See also
Breakpoints	<p>The simulation executes the model step-by-step, enabling you to validate the logic of your behavior model; the simulation halts when it reaches an element defined as a breakpoint.</p> <p>The UML elements that can be defined as breakpoints include: <i>Actions, Activities, States and most other behavioral nodes (decision, initial, final etc.)</i>.</p> <p>The UML relationships that can be defined as breakpoints include: <i>Interaction Messages</i>.</p> <p>The breakpoints are stored as Breakpoint Sets for a given Enterprise Architect project.</p> <p>Elements that are included in a simulation and that have breakpoints are marked by a green circle off the top left corner of the element, whilst the simulation is in progress. If the simulation is not running, the green circles are not displayed.</p> <p>When Javascript is enabled, all Simulation variables will be displayed in the Locals window - and it is possible to modify Simulation variables using the Simulation window's console input field (underneath the Toolbar).</p>	<p>Run Model Simulation ^[2112]</p> <p>Simulation Windows ^[2106]</p>

Toolbar Options

Icon	Action	See also
	Enables all breakpoints defined in the current Breakpoint Set for the simulation session.	
	Deletes all breakpoints defined in the current Breakpoint Set for the simulation session.	

Icon	Action	See also
	Disables all breakpoints defined in the current Breakpoint Set for the simulation session.	
	Adds a breakpoint for the selected element or Sequence message to the current Breakpoint Set.	
	Changes the selected Breakpoint Set for use in the simulation session.	
	<p>Performs Breakpoint Set commands:</p> <ul style="list-style-type: none"> • New Set: Create a new Breakpoint Set • Save As Set: Saves the current Breakpoint Set under a new name • Delete Selected Set: Deletes the current Breakpoint Set • Delete All Sets: Deletes all Breakpoint Sets saved for the diagram 	

Learn more

- [Simulation Windows](#) ^[2106]
- [Run Model Simulation](#) ^[2112]

14.8 Dynamic Simulation with Javascript

The Corporate and above editions of Enterprise Architect have the capability of using **Javascript** to **evaluate guards, effects** and other pieces of **behavior** within the Simulation context. This provides for a **fully automated, intelligent execution** of your state or activity model, with fine control over breakpoints, execution speed, simulation variables etc.

Important All Simulation level variables that will be referred to or evaluated at execution time must **be prefixed with "sim." or with "this."**; for example:

- sim.logger
- sim.Customer.name
- this.count
- this.Account.amount

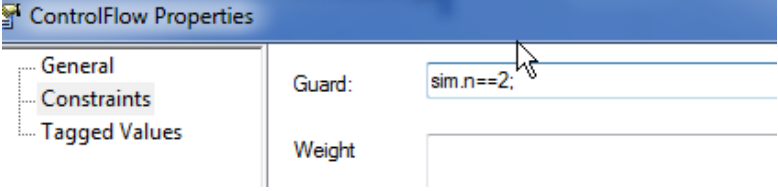
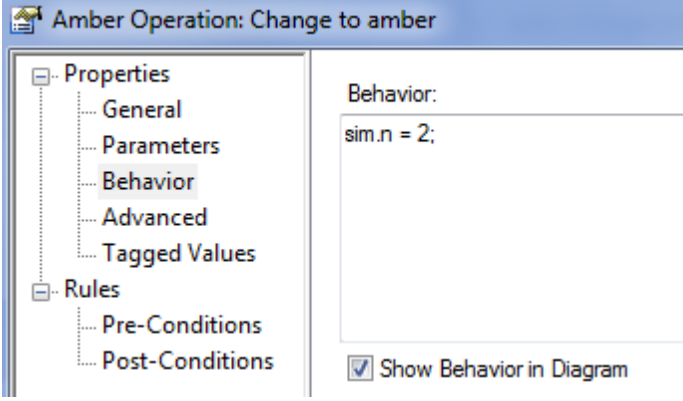
All the above are valid Simulator variables. These variables are shown in the **Local Variables** window without the **"sim."** prefix.

You would generally use the **"sim."** prefix to reference any global or control variables not declared in the owning class. You would generally use the **"this."** prefix to reference attributes of the owning class.

Some examples of where and how you can set Simulation behavior using Javascript are shown below. See the example model that comes with Enterprise Architect for some more examples. Also see the Learning Center for further information on setting up and working with Javascript in Simulations.

Using Javascript

Setting	Action	See also
Analyzer Script Input	<p>If you enter Javascript code into the Execution Analyzer window "Input" field, this code will be injected into the Simulation and executed before the Simulation starts. This is useful for establishing COM variables, global counters, useful functions and other initialization.</p> <p>Platform: UML Basic</p> <p>Options:</p> <p><input checked="" type="checkbox"/> Evaluate Guards and Effects using JavaScript</p> <p>Input</p> <pre>sim.n=1;</pre>	
Transition and Control Flow Guards	<p>This is the workhorse of the Simulation functionality. As Enterprise Architect evaluates possible paths forward at each node in a Simulation, it tests the Guards on outgoing transitions and control flows and will only move forward if there is a single true path to follow - otherwise the Simulation is considered "blocked" and manual</p>	

Setting	Action	See also
	<p>intervention is required. nb. Make sure you use the "==" operator to test for equality.</p> 	
<p>Element Behavior</p>	<p>Entry and Exit behavior may be defined for states. Such code will execute at the appropriate time and is useful for updating Simulation variables and making other assignments.</p> 	
<p>Using COM</p>	<p>One very important feature of the implementation of Javascript in Enterprise Architect's simulator is that it supports the creation of COM objects. This provides the ability to connect the running Simulation with almost any other local or remote process and either influence the Simulation based on external data, or potentially change data or behavior in the external world based on the current Simulation state (eg. update a mechanical model or software simulation external to Enterprise Architect). The syntax for creating COM objects is shown below.</p> <pre data-bbox="443 1653 1042 1765"> this.name="Odd Even"; var logger = new COMObject("MySim.Logger"); logger.Show(); logger.Log("Simulation started"); </pre>	
<p>Signalled Actions</p>	<p>It is possible to raise a signalled event (trigger) directly using Javascript. The BroadcastSignal() command is used to raise a named trigger which may influence the current state of a simulation. For example the following fragment raises the signal (trigger) named "CancelPressed".</p>	

Setting	Action	See also
	<pre style="text-align: center;">Broadcast Signal (" Cancel Pressed");</pre> <p>Note that a trigger named "CancelPressed" must exist within the current simulation environment and must uniquely have that name.</p>	
IS_IN()	<p>The IS_IN(state) operator returns true if the current simulation has an active state in any thread matching the passed in name. For example it is possible to write code such as the following to conditionally control execution:</p> <pre style="text-align: center;">if (IS_IN(" WaitingFor Input ") Broadcast Signal (" Cancel Pressed")</pre> <p>Note that the name must be unique within all contexts.</p>	
Trace()	<p>The Trace(statement) method allows you to print out trace statements at any arbitrary point in your simulation. This is an excellent means of supplementing the Simulation log with additional output information during execution.</p>	

14.9 Guards and Effects

Guards and Effects are used to **control the flow** of the simulation and to **execute additional actions** or effects during the course of a simulation.

Guards and Effects

Topic	Detail	See also
Guards	<p>Guards are conditional statements that are evaluated anytime the simulator needs to decide which path to take next. Guards typically have the following characteristics:</p> <ul style="list-style-type: none"> • Defined on transitions and control flows to govern how simulation proceeds • Written in Javascript • May refer to variables defined during simulation 	
Adding Guards	<p>Guards are defined on the transition or control flow in the properties dialog for the connector of interest. A Guard is typically a piece of Javascript that will evaluate to either true or false. For example the following is a conditional statement that refers to a current variable (Balance) being greater than zero. Note the use of the "this" prefix to indicate the variable is a member of the current class context.</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Guard: <input type="text" value="this.Balance >0;"/></p> <p>Effect: <input type="checkbox"/> Effect is a Behavior <input type="text" value=""/> <input type="button" value="..."/></p> </div>	
Evaluation Semantics	<p>During execution the Simulator will examine all possible paths forward and evaluate any guard conditions. Based on this evaluation the following may occur:</p> <ul style="list-style-type: none"> • A single valid path forward evaluates to true. The simulator will follow that path. • Two valid paths are found. The simulator will block waiting for some manual input via the console window to resolve the deadlock. • No valid path exists. Same response as when two paths are found - wait for the user to change the execution context using the console window. • No paths evaluate to true but a default (unguarded path) exists. The Simulator will take the unguarded path. 	

Topic	Detail	See also
<p>Effects</p>	<p>Effects are pieces of behavior that are executed at special times.</p> <ul style="list-style-type: none"> • On entry to a state • On exit from a state • When transitioning from one state to another (transition effect) <p>Effects may either be a piece of Javascript code or a call to another Behavior element in the current simulation.</p>	
<p>Javascript Effects</p>	<p>A Javascript effect may look like the example below, in which the Balance variable is decremented:</p> <p>Guard: <input type="text"/></p> <p><input type="checkbox"/> Effect is a Behavior</p> <p>Effect: <input type="text" value="this.Balance--;"/></p>	
<p>Call Behavior Effects</p>	<p>In the example below the effect is a call behavior effect. In this case it calls into a model Activity named "Decrement Balance" that is defined elsewhere. The simulation will then enter into that diagram/behavior and continue to execute until returning to the point at which the effect was invoked.</p> <p>Guard: <input type="text"/></p> <p><input checked="" type="checkbox"/> Effect is a Behavior</p> <p>Effect: <input type="text" value="Decrement Balance"/></p>	
<p>Order of Execution of Effects</p>	<p>In complex simulations which may involve transitioning out of deeply nested states into other deeply nested states in a different parent context, it is important to consider the following rules concerning the order of execution:</p> <ul style="list-style-type: none"> • All exit actions (effects) encountered leaving a nested context are executed in order of most deeply nested to least deeply nested • All actions (effects) defined on transitions are executed next • Finally all entry effects are executed from the least deeply nested context to the most deeply nested. <p>So the basic rule is all exit actions followed by all transition actions and finally all entry actions.</p>	

Topic	Detail	See also
Note on Javascript Variables	<p>Javascript variables to be accessed and referred to during Simulation execution must be prefixed with either:</p> <ul style="list-style-type: none">• <code>sim.</code> eg. <code>sim.pedestrianwaiting</code> - typically used for global simulation variables• <code>this.</code> e.g. <code>this.CustomerNumber</code> - typically used to refer to owning class attributes <p>This is important to let the Javascript engine know you are referring to a Simulation variable and not a simple local variable used during basic calculations etc. You can create Simulation variables of arbitrary scope and depth - eg. <code>this.customer.name</code> is a legitimate qualified name.</p>	

14.10 Triggers

Triggers represent signals and events which may **activate transitions** leaving the current state(s). A trigger may represent a **real world signal** or event such as:



- A **button** being pressed
- A **message** being received
- A **pedal** being depressed
- A **switch** being thrown
- A **state** in a concurrent region being entered or exited







For a trigger to have an effect

- Transitions have to be defined which **will fire** when the simulation **receives** the **signal/event**
- The current simulation state(s) or its parent(s) must have an **outgoing transition** that **accepts** that **trigger**
- The transition activated must be **unguarded** or have a **guard** which will **evaluate** to true

Triggers

Topic	Detail	See also						
Creating Triggers	<p>Triggers are either created as an instance of a Signal element or as an anonymous event. Triggers are connected to Transitions in the Transition Properties dialog as shown below. In this example a Trigger named 'Pushdown' has been defined based on the Signal 'Signal_Pushdown'.</p> <ul style="list-style-type: none"> • Omitting the Type and Specification details results in a simple anonymous Trigger. • If parameters are needed, these are defined on the Signal and must be supplied at the time the event fires <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p>Triggers</p> <p>Name: <input type="text" value="Pushdown"/> ...</p> <p>Type: <input type="text" value="Signal"/> Specification: <input type="text" value="Signal_Pushdown"/> ...</p> <p style="text-align: center;"> <input type="button" value="New"/> <input type="button" value="Save"/> <input type="button" value="Delete"/> </p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Specification</th> </tr> </thead> <tbody> <tr> <td>Pushdown</td> <td>Signal</td> <td>Signal_Pushdown</td> </tr> </tbody> </table> </div> <p>A trigger will appear in the Project Browser like the image below:</p> <div style="margin-top: 10px;"> <pre> graph TD Root[] --- Folder[] Folder --- Pushdown[Pushdown] Folder --- Pushup[Pushup] Folder --- Signal_Pushdown[Signal_Pushdown] </pre> </div>	Name	Type	Specification	Pushdown	Signal	Signal_Pushdown	<p>Creating Triggers ^[1173]</p> <p>Transition ^[1260]</p> <p>Simulation Events ^[2125]</p>
Name	Type	Specification						
Pushdown	Signal	Signal_Pushdown						

Topic	Detail	See also
Using Triggers	<p>Triggers are deployed by connecting them to transitions as in the example above and used during simulation by 'firing' them into the running simulation as required.</p> <p>When using triggers the following should be taken into account:</p> <ul style="list-style-type: none"> • A 'triggered' transition can not take place until its effective trigger is signalled or fired. • When a trigger is received it will activate all current waiting transitions dependent on that trigger (i.e. the trigger is broadcast) • Triggers are evaluated on all transitions for all parents of a current child state. This allows a parent state to exit all child states if necessary. • Once used in a simulation, a trigger is consumed and must be re-fired if needed again. • Sets of triggers may be saved and either manually or automatically fired to facilitate automated model simulation under different event models. 	Re-Signal Triggers ^[2130] Type-Specific Menu Section ^[938]
Firing Triggers	<p>Firing triggers means to signal or activate a trigger within the current simulation. This may activate zero, one or many waiting transitions depending on the state and concurrency of the current simulation.</p> <p>Firing triggers can be achieved in many ways. The most efficient is the Waiting Triggers list.</p> <p>During the course of model simulation, if the simulator reaches an impasse due to required triggers not being available (fired), the list of all possible candidate triggers is shown in the 'Waiting Triggers' list of the Simulation Events window (see below). Double clicking a trigger in this list will fire it into the simulation.</p> <div data-bbox="470 1637 896 1794" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <p style="text-align: center; background-color: #f0f0f0; margin: 0;">Waiting Triggers</p> <ul style="list-style-type: none">  Hold  Pushdown </div> <p>Other ways to fire a trigger include:</p> <ol style="list-style-type: none"> 1. Double click an un-signalled trigger in the Events window 	Simulation Events ^[2125] Type-Specific Menu Section ^[938]

Topic	Detail	See also															
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Sequence</th> <th style="width: 20%;">Trigger</th> <th style="width: 20%;">Status</th> <th style="width: 15%;">Type</th> <th style="width: 30%;">Parameters</th> </tr> </thead> <tbody> <tr> <td></td> <td>Pushdown</td> <td>not signalled</td> <td>Signal</td> <td></td> </tr> <tr> <td></td> <td>Pushup</td> <td>not signalled</td> <td>Simple</td> <td></td> </tr> </tbody> </table> <p>Note: You may also use the context menu on the events above to either signal an un-signalled event, or to re-signal an event which has already been fired previously.</p> <p>2. Use the context menu of the Transition required to fire and select the 'Signal Trigger in Simulation' menu option.</p>	Sequence	Trigger	Status	Type	Parameters		Pushdown	not signalled	Signal			Pushup	not signalled	Simple		
Sequence	Trigger	Status	Type	Parameters													
	Pushdown	not signalled	Signal														
	Pushup	not signalled	Simple														

14.11 Simulation Events Window

The **Simulation Events** tab of the **Breakpoints & Events** docked window is where you **manage triggers** and sets of events in a simulation. Its main functions are to -

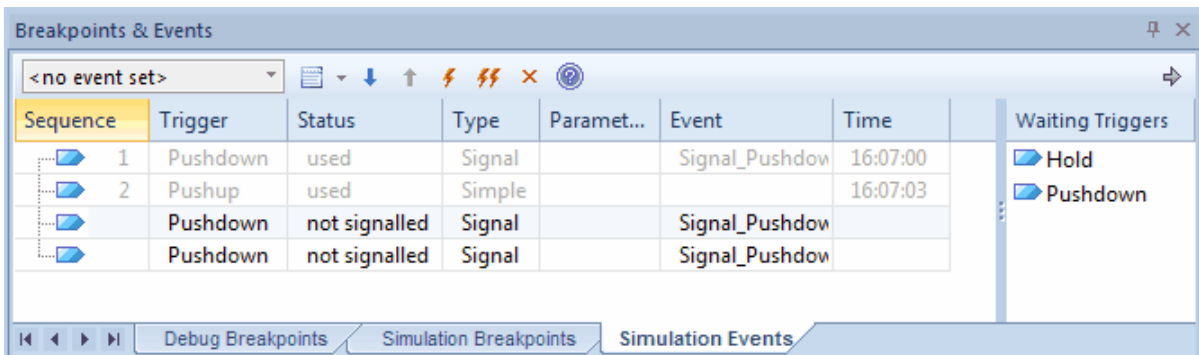
- **Add, delete and re-sequence** a set of triggers for a simulation
- **Display a list of fired, lost and waiting** events for the current running simulation
- Provide options to **fire any arbitrary trigger** into the current simulation
- Provides a convenient **'Waiting Triggers'** list of triggers that the simulation is waiting on
- Ability to **save trigger sets** for later use in both manual and automated simulations
- **Accept triggers dragged** from the Project Browser into the current list
- Enter **trigger parameters** for a waiting trigger prior to firing.

As triggers are consumed in the simulation, their status and position is logged in the main body of the Simulation Events window.

You can **save the log of fired triggers** as a trigger set or event set to reapply in another Simulation run, which you can execute manually or automatically. See the topic [Trigger Sets and Auto-Firing](#)^[2133] for more information on building and using Trigger sets.

Access [Analyzer | Breakpoints & Events > Simulation Events](#)

The following image illustrates the Simulation Events window during execution









Column Details


Field/Column	Action	See also
Sequence	During and after the simulation, indicates the position in the sequence in which a trigger was fired or is expected to be fired. Note that if a trigger is fired out of sequence, it will be moved to the bottom of the signalled events section.	

Field/Column	Action	See also
Trigger	The name of the trigger - identifies the Trigger used to initiate the event.	Triggers ^[2122]
Status	Indicates the status of the Trigger. Values can be: <ul style="list-style-type: none"> • used - the trigger has been fired and processing has passed on • lost - the trigger has been fired in the list, but it had no effect • signalled - a trigger was fired and consumed by one or more transitions • not signalled - the trigger has not yet been fired 	
Type	Indicates the type of trigger. Currently only the following are supported: <ul style="list-style-type: none"> • Signal • (no type) an anonymous trigger 	Trigger ^[1173]
Parameters	For a Signal Trigger, initially shows the parameters required for firing by the Signal specification. For example a "Login" signal may include username and password parameters - and each triggered invocation may use different parameters. Each time the simulation fires the trigger, the system will prompt you for values. You can also edit the values directly in the list when the trigger is set to not signalled . Parameters are very useful for testing the conditional logic in your simulation and to simulate a variety of inputs and data coming in from outside the simulation.	Trigger Parameters ^[2131]
Event	For a: <ul style="list-style-type: none"> • Signal Trigger, identifies the Signal specification. • For anonymous Triggers has no value 	
Time	The simulation time at which the trigger was signalled. Note that this is an absolute (real world) time, and not a relative simulation event time.	
Waiting Triggers	Lists the Triggers available for selection from the current state(s). Double-click on a trigger to add and signal it as the next trigger in the current event sequence. Note: You can show and hide this panel by clicking on the gray arrow just above the panel.	Waiting Triggers ^[2129]

Field/Column	Action	See also

Toolbar Options

Option	Action	See also
	<p>Use this drop list to select and work with previously defined trigger sets.</p> <p>Before running a simulation, select a previously-defined trigger set to use for the next simulation run. You elect to not use a trigger set by selecting the <no event set> option.</p>	Trigger Sets and Auto-Firing ^[2133]
	<p>Click to create and delete trigger sets:</p> <ul style="list-style-type: none"> • Save Set - Save the current trigger list as a new trigger set; the system prompts you for a name for the new set • Save Set As - Create a copy of the current set under a new set name • Delete Selected Set - Delete the current trigger set • Delete All Sets for Diagram - Delete all saved trigger sets for the current diagram 	Trigger Sets and Auto-Firing ^[2133]
	<p>Move the selected trigger one line down in the firing sequence of triggers.</p> <p>This option is not available if there are no not signalled triggers below the selected line.</p>	
	<p>Move the selected trigger entry one line up in the firing sequence of triggers.</p> <p>This option is not available if there are no not signalled triggers above the selected line.</p>	
	<p>Click to fire the selected trigger. You can also fire the trigger by double-clicking on it.</p>	
	<p>Click to toggle auto-firing on and off.</p> <p>Auto-firing will fire the un-signalled triggers in your trigger set sequentially. If your set matches a valid execution path, then the simulation will run automatically. Out of sequence or unused triggers will be 'lost'.</p> <p>A breakpoint pauses the auto-firing and you will need to click on the next trigger to resume auto-firing the simulation.</p>	Trigger Sets and Auto-Firing ^[2133]

Option	Action	See also
	Delete the selected trigger(s) from the list.	

Context Menu Options

Options	Action	See also
Signal Selected	Signal, or fire, the selected not signalled trigger.	
Remove Selected	Remove a not signalled trigger from the sequence.	
Re-Signal Selected	Fire a used or signalled trigger again.	
Set All to Unsignalled	Set all used or signalled triggers to not signalled .	
Clear Trigger List	Clear all triggers from the window, regardless of their status.	

Learn more

- [Type-Specific Menu Section](#) ⁹³⁸

14.12 Waiting Triggers

When a simulation reaches a point where any change of state (for any thread) **requires a Trigger to proceed**, the simulation is effectively **paused** and control returns to Enterprise Architect. The simulation is now effectively waiting for some form of event (real world signal) to proceed. The **Waiting Triggers list** is now useful in helping to determine which Trigger should be manually signalled.

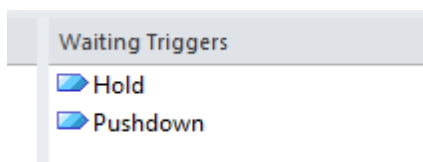
Access [Analyzer | Breakpoints & Events > Simulation Events](#)

The **Waiting Triggers** list on the **Simulation Events** tab is

- **Populated on each Simulation cycle** with any Triggers that would have an immediate effect if signalled
- Populated with a **discrete set** (any duplicates are not shown as a Trigger is effectively **broadcast** to all transitions at once)
- Activated by **double clicking** on the Trigger of interest
- Includes **all possible triggers** - including those activating transitions on **parents** of currently nested states

The example below shows the current simulation has hit a point where two possible Triggers may influence the flow of execution. Due to the nature of Triggers and their effects, the list below may refer to the following example situations equally validly:

- A single state has two outgoing transitions which are respectively waiting for Hold and Pushdown. Firing one of these will activate the associated transition in the simulation
- Two (or more) threads (concurrent regions) each have a state waiting on either Hold or Pushdown. Firing one of these triggers will result in the thread(s) waiting on that trigger to proceed while the other thread(s) will remain blocked.
- A child state is waiting on one of the triggers while a parent state is waiting on the other. Firing a trigger will result in the associated transition being fired and either the child or parent proceeding respectively.
- Any combination of the above



Learn more

- [Type-Specific Menu Section](#) ⁹³⁸

14.13 Re-Signal Triggers

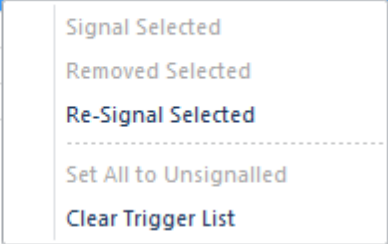
It is possible to **re-signal a Trigger** as a shortcut of dragging in additional Trigger instances for signalling.

Access Analyzer | Breakpoints & Events > Simulation Events

The Simulation Events window contains a list of Triggers which have **already fired**. By **right clicking on a Trigger** that you wish to signal again, you can use the context menu to cause the **re-signal** to happen.

The image below demonstrates this in action. When a signal is re-signalled, a new copy is made and placed at the end of the signalled triggers list, where it is automatically fired again.

Sequence	Trigger	Status	Type	Parameters
1	Pushdown	used	Signal	
2	Pushup	used		
3	Pushdown	used		
4	Pushup	used		



A context menu is displayed over the table, listing the following options: Signal Selected, Removed Selected, Re-Signal Selected (highlighted), Set All to Unsignalled, and Clear Trigger List.

14.14 Trigger Parameters

Trigger parameters are **arguments passed into the simulation along with a trigger** when it is fired. They allow for **complex behavior** to be specified decision to be made based on variables and data passed into a simulation at run-time by a fired trigger (event) .

Topic	Detail										
Introduction	<p>To use trigger parameters you</p> <ul style="list-style-type: none"> • First create a Signal element with the appropriate attributes that will become your parameters at run time • On a suitable transition in your diagram, create a trigger that is based on the signal created earlier • Now at run-time you will be prompted to enter suitable parameters - they are then passed in along with the trigger. 										
Signals	<p>A Signal element is a template or specification from which actual triggers can be built. The example below has two arguments, a Name and a Password. These will be filled in at execution time either manually or as part of a pre-defined trigger set.</p> <div data-bbox="445 1182 826 1424" style="border: 1px solid black; padding: 10px; background-color: #fff9c4; margin: 10px auto; width: fit-content;"> <pre> «signal» Login - Name :String - Password :String </pre> </div>										
	<p>The Trigger parameters 'prompt' that asks for suitable values for each parameter. Note that you need to enclose strings in double quotes, otherwise the interpreter will think you are referring to other variables.</p> <div data-bbox="445 1641 1110 1872" style="border: 1px solid #ccc; padding: 10px; background-color: #e0e0e0; margin: 10px auto; width: fit-content;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="background-color: #4a86e8; color: white;">Trigger Parameter Entry</th> </tr> <tr> <th style="width: 70%;">Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td colspan="2" style="background-color: #e0e0e0;">Login</td> </tr> <tr> <td>Name</td> <td>"Joe"</td> </tr> <tr> <td>Password</td> <td>"secret"</td> </tr> </tbody> </table> </div>	Trigger Parameter Entry		Property	Value	Login		Name	"Joe"	Password	"secret"
Trigger Parameter Entry											
Property	Value										
Login											
Name	"Joe"										
Password	"secret"										

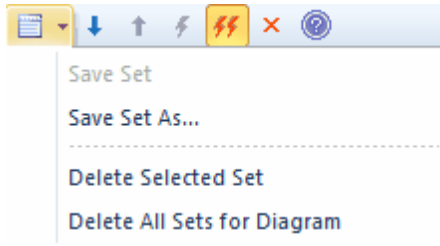
Topic	Detail
	<p>Below is an example diagram that makes use of trigger parameters. At the Evaluate Login state, the simulation examines the variables passed in as trigger parameters and makes a decision to either accept the credentials or deny them.</p> <pre> stateDiagram-v2 [*] --> Login : Initial Login --> EvaluateLogin : Login state Login { /sim.LoginCount =0; } state EvaluateLogin { [this.Login.Name=="Joe" && this.Login.Password=="secret";] } EvaluateLogin --> OnLine : Choice EvaluateLogin --> Fail : Choice state OnLine { Final } state Fail { + entry / OnEnter } Fail --> Login : Retry Fail --> Offline : Choice state Offline { [sim.LoginCount>3;] } Offline --> Final </pre>

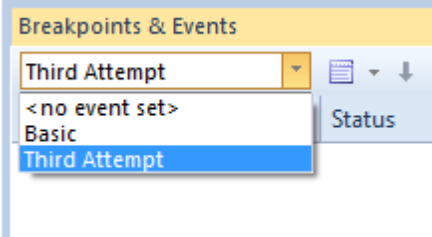
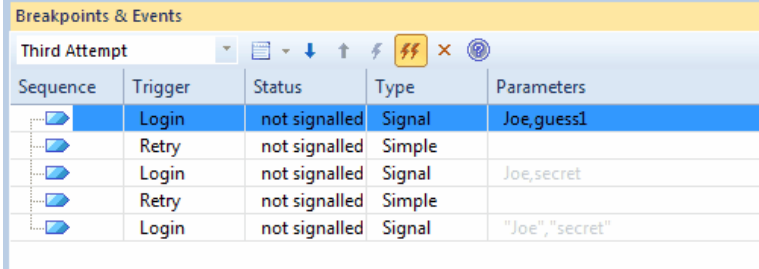

14.15 Trigger Sets and Auto-Firing

Trigger Sets are a powerful means of **automating and streamlining the execution**, testing and validation of simulation models. By **re-using sets of triggers** (with or without parameters) it is possible to quickly and efficiently walk through many simulation scenarios, either manually or automatically using the 'auto-firing' tool.

Access [Analyzer | Breakpoints & Events > Simulation Breakpoints](#)

About Trigger Sets

Topic	Detail	See also
Trigger Sets	<ul style="list-style-type: none"> • Stored with an associated diagram • Made up of a list of Triggers in a set sequence • May include Trigger parameters where necessary • May be used manually by double clicking Triggers to fire as required • May be used as part of the "auto-fire" behavior to automate execution • Managed from the Simulation Events Window 	
Managing Sets	<p>Trigger sets can be created by manually dragging triggers into the active triggers list and then using the Manage Trigger Sets drop menu to save a new set.</p> <p>It is also possible to save a set of triggers built up during a single simulation setting as a new set. This is convenient for creating multiple test paths through a simulation, based on saving the manually fired triggers for each test case.</p> <p>The Manage Triggers Menu</p>  <p>You can also delete a set and delete all sets for the current diagram.</p> <p>It is also possible to load a set, modify parameters and/or order of firing and save the set with a new name. This is a convenient method for rapidly creating a suite of simulation test scripts.</p>	

Topic	Detail	See also
<p>Using Sets</p>	<p>To use a trigger set you first select it by name from the trigger set drop list as in the example image below. Once selected it loads the trigger list window with the defined trigger set.</p> <p>Note that the special item <no event set> means no set is currently selected. At the start of each simulation, if a set is selected, it will be loaded afresh for the next run. If <no event set> is selected, the trigger list will be cleared.</p>  <p>Once you have selected a trigger set and the list of triggers loaded you have two options:</p> <ul style="list-style-type: none"> • Fire the triggers as required manually • Use the auto-fire feature to fully automate the simulation (see below) 	
<p>Auto-Firing</p>	<p>Auto-firing is a convenient way of streamlining your simulations. Once you have loaded a trigger set, if you select the auto-fire button (below) then Enterprise Architect will automatically pick up waiting triggers when it reaches an impasse in the simulation. In practice, this means that trigger sets matching exactly a path through the simulation with automatically run without your intervention.</p> <p>As you can save any number of trigger sets with different pathways and trigger parameters, you can effectively and quickly test and work with many different scenarios.</p> 	
<p>Auto-Firing Rules</p>	<p>When a simulation runs with auto-firing enabled, Enterprise Architect will wait until a point is reached where the simulation is 'blocked' or</p>	

Topic	Detail	See also
	<p>stable, waiting on one or more triggers to advance the simulation. At that time, the first unfired trigger in the list will be picked up and fired into the simulation. The outcome depends on the relevance and perhaps on the parameters of the trigger.</p> <ul style="list-style-type: none"><li data-bbox="491 461 1153 533">• If the trigger matches a 'waiting' trigger it is immediately consumed and the simulation advances<li data-bbox="491 555 1225 757">• If the trigger matches no 'waiting' trigger or possible parent transition, then the trigger is 'lost' and the simulation remains in the current state. This corresponds to a scenario such as a user pressing an 'on' button several times in succession - there is no effect other than that occasioned by the first press.	

14.16 Using Trigger Sets to Simulate an Event Sequence

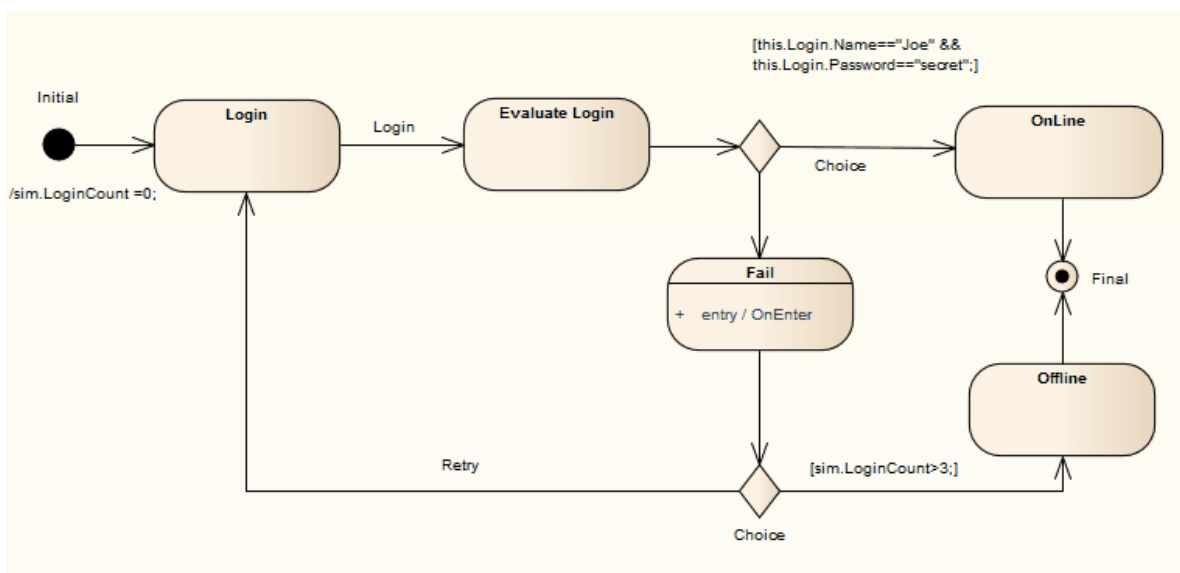
As a simple example of how useful **trigger sets** are, consider the following trigger set and associated diagram.

In this example we **simulate** a simple "three strikes and you are out" login process, taking a user name and password. The success path is waiting for the name "Joe" and the password "secret" (note - it is very important that parameters referencing strings are enclosed in quotes - otherwise the interpreter thinks the name refers to another variable within the simulation.

- Pass 1 tries Joe and guess1 - which fails
- Pass 2 tries Joe and secret, but as they are referring to variables, not strings - this fails as well
- Pass 3 shows the correct way of referencing trigger parameters and the simulation will succeed

Breakpoints & Events				
Third Attempt				
Sequence	Trigger	Status	Type	Parameters
1	Login	not signalled	Signal	Joe,guess1
2	Retry	not signalled	Simple	
3	Login	not signalled	Signal	Joe,secret
4	Retry	not signalled	Simple	
5	Login	not signalled	Signal	"Joe","secret"

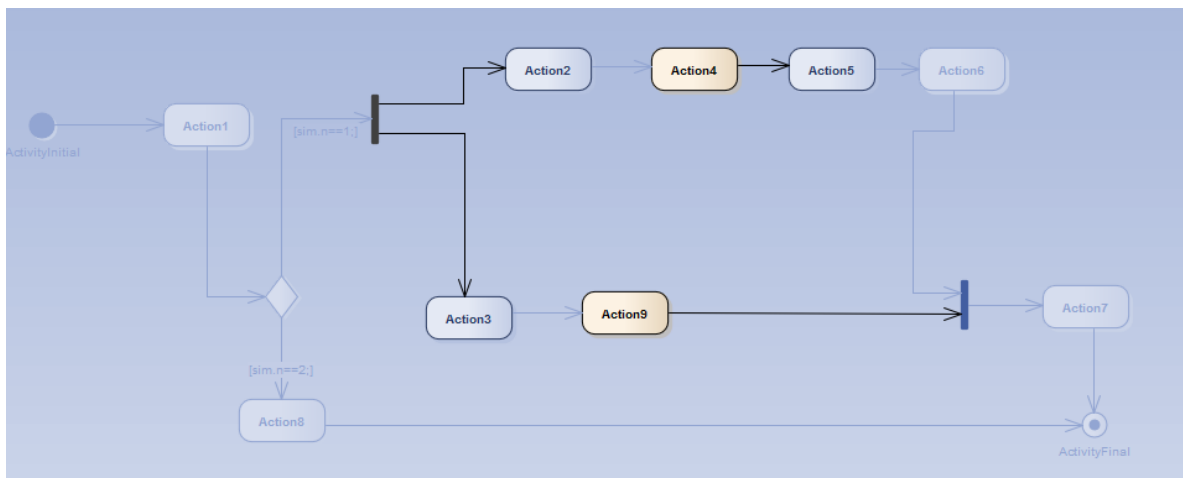
The simple diagram simulating a login process **requiring a username and password pair**.



14.17 Multi-threading - Forks and Joins

The Model Simulator includes the ability to handle **multi-threaded simulations** using **Fork** and **Join** nodes.

- In the example below, the current **execution point** has forked into **two threads**, each with its own active node.
- As this example progresses, the lower branch will **wait** at the **Join** node until the top branch has completed all its Actions.
- Once the two threads **merge back into one**, the Simulation will continue as a **single thread** until completion.
- When automatically stepping, each thread will be seen to execute a single step during one simulation "cycle" - although when single stepping or at a breakpoint, the behavior is to alternate step between threads as each thread receives processing time.
- Note that the **Call Stack** window will show **two active threads** and one "paused" thread in the example below. Once the threads merge there will be a return to single threaded execution.
- Also note that the **Local variables are shared (global) between all threads**. If you want to Simulate private variables on a thread you must create new Simulation variables at the start of each thread - pre-loading such variables with existing global data.



Learn more

- [Fork/Join](#)^[1130]

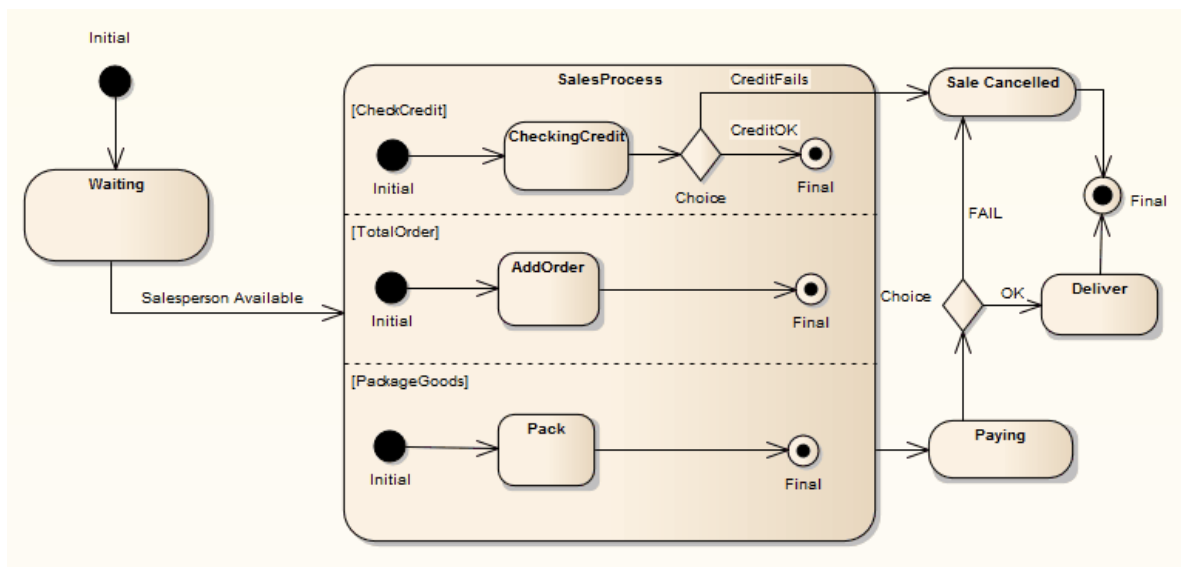
14.18 Multi-threading - Concurrent State Regions

Concurrent regions within a state represent state changes and processing that occurs in **parallel** inside one overall parent state. This is especially useful when **one region raises events** or modifies simulation variables that another region is dependent on. For example, one region could contain a simulated timer which raised events on set intervals that invoked state changes in the states within other regions.

Concurrent regions are essentially the same as Forks and Joins with similar logic and processing rules.

In the example below,

- When the transition to SalesProcess is taken, each region is **concurrently activated**.
- Credit is checked, the order totalled and the goods required packed up.
- However, in the event that the CreditCheck fails, this triggers the transition to the Sale Cancelled state. Note that when this occurs, the entire parent state and all owned regions are immediately exited, regardless of their processing state.
- If the Credit Check succeeds, the region moves to the final state and once the other regions have all reached their own final state, the parent state can then be exited.





14.19 BPMN Simulation

BPMN simulation is a method for validating the behavior of your **BPMN Business Process** diagrams. With visual indications of all currently executing activities along with the possible activities that can be executed next you will easily be able to identify potential issues with the process you have modeled.




In Enterprise Architect you will need the Business & Software Engineering or Ultimate edition in order to use BPMN simulation.

Working with BPMN Simulation

Topic	Detail	See also
Creating a BPMN Simulation Model	This topic describes any special steps that are required to create a BPMN model that is suitable for simulation.	Creating a BPMN Simulation Model  [2139]
Comparing UML Activities to BPMN Processes	This topic provides an overview of the differences between UML Activity diagrams and BPMN Business Process diagrams.	Comparing UML Activities to BPMN Processes  [2140]

14.19.1 Creating a BPMN Simulation Model

To create a BPMN diagram suitable for simulation

Step	Action	See also
1	Create a Business Process or BPEL diagram from the BPMN 2.0 technology. If you create a BPEL diagram Enterprise Architect has specialized dialogs to streamline the creation of compliant models.	Add New Diagrams  [724]
2	We recommend creating a Start Event to clearly show where your simulation starts. You will be given multiple choices for the Event Type , these do not influence the simulation of your model. If no Start Events are defined then the simulation will start from an Activity that has no incoming Sequence Flows .	Start Event  [1601]
3	Add all of the Activities that are involved in the Process being modeled. You will be given multiple choices for the Task Type , these do not influence the simulation of your model. The behavior of Activities can be further decomposed by specifying an Activity Type of Sub-Process and selecting Embedded or CallActivity . Standard Loops are also supported.	Activity  [1605]

Step	Action	See also
4	Add Sequence Flows between your activities. The BPEL properties dialog allows you to enter the condition that must be true before the Sequence Flow will be followed. You can also set the Condition Type to Default to ensure that this flow will be taken if all other branches fail the condition specified. If you are not using a BPEL diagram, please refer to the conditionExpression and conditionType tagged values.	Sequence Flow ^[1612]
5	Add End Events for any conditions that will cause the process or active execution path to end. You will be given multiple choices for the Event Type , of these only the Terminate type will influence the execution. In simulations with multiple active nodes, it causes the entire process to terminate instead of just the thread that reaches that node.	End Event ^[1608]
6	You may optionally define a simulation script for the execution of this diagram. Ensure you select BPMN as the Platform .	Set Up Simulation Script ^[2109]

Learn more

- [BPMN 2.0 Toolbox Pages](#)^[1549]
- [Create BPEL 2.0 Model Structure](#)^[1597]

14.19.2 Comparing UML Activities to BPMN Processes

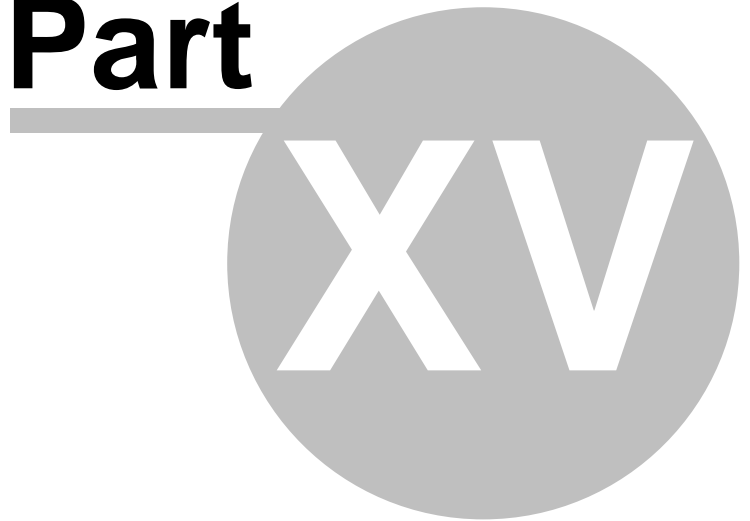
The execution and simulation of BPMN models provide a number of similarities to UML Activity diagrams. The following table provides a mapping between similar concepts and the differences between the two methods to expressing the behavior of a system.

Comparing UML and BPMN Simulation

UML Activity	BPMN Business Process	See also
The starting point is defined by an Initial Node . No method of specifying why the Activity was started is available.	The starting point is defined by a Start Event . This implies a specific cause for the Activity to start, although it may be unspecified.	Initial Node ^[1136] Start Event ^[1601]
The basic behavior unit in an Activity is the Action element. UML provides many different forms of Actions, although the simulation makes use of a small subset of these.	The basic behavior unit in an Activity is the Activity element. A number of different Task Types are available. These typically describe different methods of execution (for example Manual) as opposed to what happens.	UML Action ^[1091] BPMN Activity ^[1605]
A Control Flow is used to connect the elements on an Activity diagram. A distinguishing feature is that only a single control flow may be followed from any node	A Sequence Flow is used to connect the elements on a Business Process diagram. These differ from UML Activity diagrams in that all valid sequence flows are taken by default. To restrict flow on a sequence flow set the Condition Type to	Control Flow ^[1219] Sequence Flow ^[1612]

UML Activity	BPMN Business Process	See also
except for an explicit Fork Node. To restrict flow on a control flow, add a guard .	expression and and the Condition Expression .	
A Decision node is used to explicitly model a decision being made. A Merge node, which uses the same syntax is used when the potential flows are combined back into one.	A Gateway node set to Exclusive is used when a single path must be selected. It is also used to combine the potential flows again. A direction may be specified as Converging or Diverging to explicitly select between the two modes.	Decision ^[1117] Gateway ^[1607]
A Fork node is used to concurrently execute multiple nodes, while a Join node, using the same syntax is used to wait for all incoming flows to become available and leave with a single flow.	A Gateway node set to Parallel is used to explicitly model concurrent execution of multiple nodes. It is also used to wait for all incoming flows to become available and leave with a single flow. A direction may be specified as Converging or Diverging to explicitly select between the two modes.	Fork/Join ^[1130] Gateway ^[1607]
There is no allowance for concurrently executing only some outputs from a node for UML Activities. If you needed this you would need to add later control flows with the appropriate guards.	A Gateway node set to Inclusive is used to explicitly model the situation where all outgoing flows with a true condition are executed concurrently.	Gateway ^[1607]
A Call Behavior Action is used when behavior needs to be further decomposed by referring to an external activity.	Activity elements are set as an CallActivity Sub-Process when behavior needs to be further decomposed by referring to an external activity.	UML Action ^[1091] BPMN Activity ^[1605]
Activity elements are used when behavior needs to be further decomposed without referring to an external activity.	Activity elements are set as an Embedded Sub-Process when behavior needs to be further decomposed without referring to an external activity.	UML Activity ^[1101] BPMN Activity ^[1605]

Part



15 Execution Analyzer



This section describes the recording and visual analysis of executing applications in Enterprise Architect.

The Visual Execution Analyzer builds on and processes the structures and operations of the Model Driven Development Environment (MDDE). The MDDE provides tools to design, build and debug an application, forming part of the Debugger facilities of Enterprise Architect.

One of the primary objectives of the Visual Execution Analysis feature is to enable you to work with the stack traces you captured in debugging your application, to generate Sequence diagrams. This is a great way to document and understand what your program is doing during its execution phase.






Access **Analyzer**
Project Manager package context menu | Execution Analyzer

Use to

- Optimize existing system resources and understand resource allocation
- Ensure that the system is following the rules as designed
- Produce high quality documentation that more accurately reflects system behavior
- Understand how and why systems work
- Train new employees in the structure and function of a system
- Provide a comprehensive understanding of how existing code works
- Identify costly or unnecessary function calls
- Illustrate interactions, data structures and important relationships within a system
- Trace problems to a specific line of code, system interaction or event
- Visualize why a sequence of events is important
- Establish the sequence of events that occur immediately prior to system failure
- Simulate the execution of behavior models including State Machines, Activities and Interactions

Topics

Topic	See
Create a sample model for your language compiler, on which to examine the facilities of the Visual Execution Analyzer.	Visual Execution Analyser Samples ^[2145]
Record executing programs and represent the behavior as a UML Sequence diagram; recording is supported for:	Recording Sequence Diagrams ^[2147]

Topic	See
<ul style="list-style-type: none"> • Microsoft Windows Native C • Microsoft Windows Native C++ • Microsoft Windows Visual Basic • Microsoft .NET Family (C#, J#, VB) • Sun Microsystems Java • PHP 	
Record native Windows applications and profile their behavior.	Profiling Native Applications 
Create and work with objects created within the Enterprise Architect modeling environment using a dynamic Object Workbench.	Object Workbench 
Run xUnit tests.	Unit Testing 
Testpoint Management - provides the facility to pass or fail application tasks, viewing test results in real time as the program executes and results are saved.	Testpoint Management 
Simulate UML behavior models to verify their logical and design correctness, for: <ul style="list-style-type: none"> • Activities • Interactions and Sequences • State Machines 	Model Simulation 

Learn more

- [Code, Build & Debug](#)  (for the Enterprise Architect MDDE and Debugger)

Learning Center topics

- **(Alt+F1) | Execution Analysis**

15.1 Visual Execution Analyzer Samples

Enterprise Architect enables you to easily import complete sample models (packages), including all necessary model information, code and build scripts. These sample patterns make it simple to explore and try out the Visual Execution Analyzer. You can generate an example model for:

- Java
- Microsoft.NET
- Microsoft C++
- PHP Apache

Access **Project | New Model (Ctrl+Shift+M) > VEA Examples**
Project Browser | Package context menu | Add | New Model > VEA Examples

Reference

Field	Usage	See also
Technology	Select the appropriate technology.	
Name	Displays the samples available for the selected technology; select the required sample to import.	
description field	Displays a description of the selected sample.	
Destination folder	Browse for and select the directory in which to load the source code for the sample.	
Use Local Path	Enable the selection of an existing local path to place the source code under; changes the Destination folder field to a drop-down selection.	
Compiler command	Displays the default compiler command path for the selected technology; you must either: <ul style="list-style-type: none"> • Confirm that the compiler can be found at this path, or • Edit the path to the compiler location 	Local Paths [1900] Local Paths Dialog [1901]
Edit Local Paths	Many VEA examples specify their compiler using a local path. The first time you use any sample you must click on this button to ensure the local path points to the correct location. The Local Paths dialog displays.	

Notes

- If required, you can define custom samples by adding files to the *AppSamples* directory where Enterprise Architect is installed; top-level directories are listed as Technologies and can contain an icon file to customize the icon displayed for the technology
 - Directories below this are defined as groups in the patterns list; the patterns are defined by the presence of four files with a matching name: a zip file (.zip), XML file (.xml), config file (.cfg) and optional icon (.ico)
- The *config* file supports the following fields:
 - [provider], [language], [platform], [url], [description], [version] - all displayed in the description field
 - [xmireootpaths] - the root path of the source code in the exported xmi; this is replaced with the selected destination folder when the user applies the application pattern

Learn more

- [Model Wizard](#)^[660]
- [Recording Sequence Diagrams](#)^[2147] }
- [Profiling Native Applications](#)^[2170] }
- [Object Workbench](#)^[2176] } Use the Visual Execution Analyzer samples to explore these
- [Unit Testing](#)^[2183] } areas of Visual Execution Analysis
- [Testpoint Management](#)^[2187] }
- [Model Simulation](#)^[2100] }

15.2 Recording Sequence Diagrams

This section explains how to use the Visual Execution Analyzer to record execution flow in the form of a Sequence diagram.

A Sequence diagram provides easy to understand visual information including:

- A representation of how information is passed through a system
- The sequence of various functions and their corresponding parameters
- A clear view of how different Classes interact to create behavior
- A visual overview of how data structures are used to produce results

A Sequence diagram extends traditional analysis to help identify errors in logic, explain unexpected system behavior and identify data flow inconsistencies; the Visual Execution Analyzer extends analysis through the use of a comprehensive array of reports that detail everything from state transitions through to the contents of the stack at a given time.

Access [Analyzer](#) | [Record](#) | [Recorder](#)

Use to

- Set up and control the recording of code execution
- Interpret the results of recording
- Generate Sequence diagrams from the recording, and moderate the contents of the diagram

Topics

Review the following topics in sequence:

Topic	Link
An overview of how the Visual Execution Analyzer creates a visual representation of the execution of an application, outlining the functions that are being called, the types of messages being sent, the key data structures used and the relationships between different Classes. The diagram makes it much simpler to understand how information is moved throughout the system and what values are being passed by various functions.	How It works ^[2148] The Recording History ^[2149] Diagram Features ^[2152]
Preparing to record execution of the application.	Setup for Recording ^[2153]
Deploying recording markers, breakpoints and marker sets.	Place Recording Markers ^[2154]
Controlling the recording session, using the Record & Analyze window.	Control the Recording Session ^[2161]
Interpreting the results of a recording, saving and retrieving these results and generating a Sequence diagram from the recording.	Generating Sequence Diagrams ^[2164]

Topic	Link
A description of how you can generate Sequence diagrams that show transitions in state as a program executes.	Reporting State Transitions ^[2165]

Notes

- Recording is available to users of all editions of Enterprise Architect except the Desktop edition

Learning Center topics

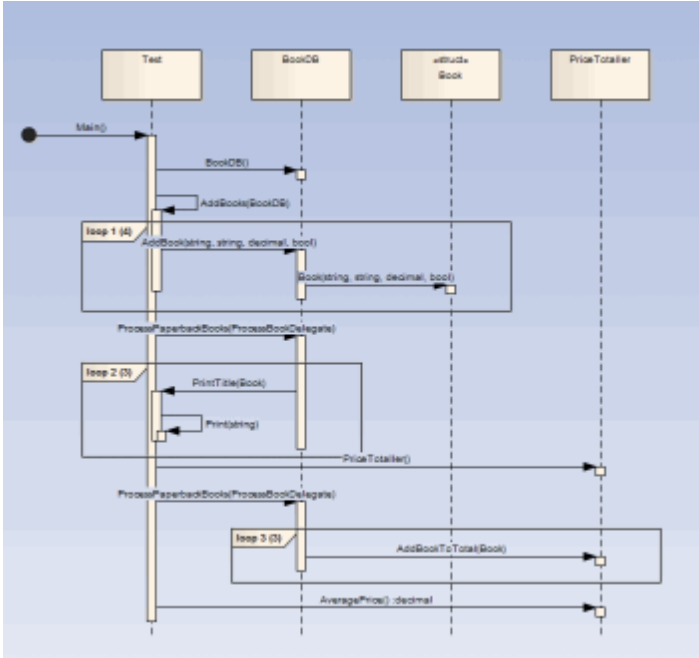
- (Alt+F1) | [Execution Analysis | C++/Native Samples | C++ Recording 1](#)
- (Alt+F1) | [Execution Analysis | C++/Native Samples | C++ Recording 2](#)
- (Alt+F1) | [Execution Analysis | C++/Native Samples | C++ Recording 3](#)

15.2.1 How it Works

This topic explains how the Visual Execution Analyzer generates Sequence diagrams.

Topics

Topic	Detail	See also
Usage	<p>The Visual Execution Analyzer enables you to generate a Sequence Diagram.</p> <p>The diagram creates a visual representation of the execution of an application, outlining what functions are being called, the types of messages being sent, the key data structures used and the relationships between different Classes; that is, how information is moved throughout the system and what values are being passed by various functions.</p> <p>The diagram below illustrates the Sequence Diagram output for a program that calculates the price of books.</p> <p>The first loop structure is executed four times, to add four books to the book database; the arrows indicate information flow and demonstrate the change of states over time.</p>	

Topic	Detail	See also
	 <p>The diagram is a UML Sequence Diagram with four lifelines: Test, BookDB, aBook, and PriceTotaler. It starts with a 'Main()' call to the Test lifeline. Test sends a 'BookDB()' message to BookDB. BookDB returns 'AddBooks(BookDB)' to Test. Test enters 'loop 1 (4)' and sends 'AddBook(string, decimal, bool)' to BookDB. BookDB returns 'Book(string, decimal, bool)' to Test. Test then sends 'ProcessPaperbackBooks(ProcessBookDelegate)' to BookDB. Test enters 'loop 2 (3)' and sends 'PrintTitle(Book)' to BookDB. BookDB returns 'Print(string)' to Test. Test then sends 'PriceTotaler()' to PriceTotaler. Test enters 'loop 3 (3)' and sends 'AddBookToTotal(Book)' to PriceTotaler. PriceTotaler returns 'AveragePrice() decimal' to Test. Finally, Test sends 'ProcessPaperbackBooks(ProcessBookDelegate)' to BookDB.</p> <p>Enterprise Architect can record arguments to functions, record calls to external modules or capture state transitions based on any given constraint; this information can be integrated with existing system knowledge and test data to optimize code execution, reduce errors and determine why application failure and system crashes occur.</p> <p>If an application crashes, data corruption such as a stack overflow can prevent you from diagnosing and rectifying the problem; however, the Visual Execution Analyzer enables you to record a given execution sequence to provide a reliable source of information that might further explain why a crash occurred.</p> <p>A Sequence diagram can convey more detail and provide greater understanding than reading unfamiliar code that might have been written by someone else; it also makes it easier to document existing code when the Sequence diagram illustrates functions that are being called and the specific sequence of events that occur to produce a particular type of system behavior.</p>	<p>The Recording History ^[2149]</p> <p>Diagram Features ^[2152]</p>

Learn more

- [Setup for Recording](#) ^[2153]

15.2.1.1 The Recording History

All information recorded when the execution analysis of an application encounters user-defined recording markers is held in the Record & Analyze window.

Topics

Topic	Detail	See also
Columns	<p>The columns in the Record & Analyze window are as follows:</p> <ul style="list-style-type: none"> • Sequence - The unique sequence number • Threads - The operating system thread ID • Delta - The elapsed thread CPU time since the start of the sequence • Method - There are two Method columns: the first shows the caller for a call or for a current frame if a return; the second shows the function called or function returning • Direction - <i>Stack Frame Movement</i>, can be <i>Call</i>, <i>Return</i>, <i>State</i>, <i>Breakpoint</i> or <i>Escape</i> (<i>Escape</i> is used internally when producing a Sequence diagram, to mark the end of an iteration) • Depth - The stack depth at the time of a call; used in the generation of Sequence diagrams • State - The state between sequences • Source - There are two Source columns: the first shows the source filename and line number of the caller for a call or, if a return, for a current frame; the second shows the source filename and line number of the function called or function returning • Instance - There are two Instance columns, which only have values when the Sequence diagram produced contains State Transitions; the values consist of two items separated by a comma - the first item is a unique number for the instance of the Class that was captured, and the second is the actual instance of the Class <p>For example: supposing a Class <i>CName</i> has an internal value of 4567 and the program created two instances of that Class; the values might be:</p> <ul style="list-style-type: none"> • 4567,1 • 4567,2 <p>The first entry shows the first instance of the Class and the second entry shows the second instance.</p> <p>You can perform a number of operations on the results of a recording, using the Record & Analyze window context menu, once the recording is complete.</p>	<p>Working With Recording History <small>[2151]</small></p>
Toolbar	<p>The Record & Analyze window toolbar provides a range of facilities for controlling the recording of the execution of an Analyzer script.</p>	<p>Recorder Toolbar <small>[2163]</small></p>

Notes

- The checkbox against each operation is used to control whether or not this call should be used to create a Sequence diagram from this history
- In addition to enabling or disabling the call using the checkbox, you can use context menu options to enable or disable an entire call, all calls to a given method, or all calls to a given Class

15.2.1.2 Working With Recording History

You can perform a number of operations on or from the results of a recording session, using the Record & Analyze window context menu.

Reference

Option	Detail	See also
Show Source for Caller	Display the source code, in the Source Code Viewer, for the method calling the sequence.	Editing Source Code ^[1799]
Show Source for Callee	Display the source code, in the Source Code Viewer, for the method being called by the sequence.	
Generate Diagram for Selected Sequence	Generate a Sequence diagram for a single sequence selected in the recording history.	Generating Sequence Diagrams ^[2164]
Generate Sequence Diagram	Generate a Sequence diagram including all sequences in the recording history.	
Clear	Clear the recording history currently displayed in the Record & Analyze window.	The Recording History ^[2149]
Save Recording History to File	Save the recording history to an XML file. A browser window displays, on which you specify the file path and name for the XML file.	
Load Recording History From File	Load a previously saved recording history from an XML file. A browser window displays, on which you specify the file path and name for the XML file to load.	
Disable All Calls	Disable every call listed in the Record & Analyze window.	

Option	Detail	See also
Disable This Call	Disable the selected call.	
Disable This Method	Disable the selected method.	
Disable This Class	Disable the selected Class.	
Disable All Calls Outside This Call	Disable every call listed in the Record & Analyze window <i>except for</i> the selected call.	
Enable All Calls	Enable every call listed in the Record & Analyze window.	
Enable This Call	Enable the selected call.	
Enable This Method	Enable the selected method.	
Enable This Class	Enable the selected Class.	
Help	Display the Help topic for the Record & Analyze window.	The Recording History ^[2149]

15.2.1.3 Diagram Features

When you generate a Sequence diagram, it includes the following features:

Feature	Detail	See also
References	When the Visual Execution Analyzer cannot match a function call to an operation within the model, it still creates the sequence but also creates a reference for any Class that it cannot locate. It does this for all languages.	Generating Sequence Diagrams ^[2164]
Fragments	Fragments displayed in the Sequence diagram represent loops or iterations of a section(s) of code. The Visual Execution Analyzer attempts to match function scope with method calls to as accurately as possible represent the execution visually.	
States	If a State Machine has been used during the recording process, any transitions in State are presented after the method call that caused the transition to occur.	Reporting State Transitions ^[2165]

Feature	Detail	See also
	States are calculated on the return of every method to its caller.	

15.2.2 Setup for Recording

This section explains how you prepare to record execution of the application.

Topic	Link
Prerequisites - To set up the environment for recording Sequence diagrams you must: <ul style="list-style-type: none"> Have completed the basic setup for Build & Debug and created Execution Analysis scripts for the package Be able to successfully debug the application 	Prerequisites ^[1825] Analyzer Scripts ^[1827] Debugging ^[1867]
Narrow the focus of a recording by applying filters.	Configuring Recording Detail ^[1863]
Control the detail of a recording by adjusting the stack depth.	Control Stack Depth ^[2153]


15.2.2.1 Control Stack Depth

When recording particularly high-level points in an application, the stack frame count can result in a lot of information being collected; to achieve a quicker and clearer picture, it is better to limit the stack depth on the toolbar of either:

- The Breakpoint and Markers window or
- The Record & Analyze window

Access [Analyzer | Record | Recorder](#)

Topics

Topic	Detail	See also
Usage	<p>Set the recording stack depth in the numerical field on the toolbar of the Breakpoints & Events window or the Record & Analyze window:</p>  <p>By default, the stack depth is set to four frames.</p> <p>The depth is relative to the stack level where the Debugger first encountered the recording marker; so, when recording begins, if the stack level is 6 and the stack depth is set to 3, the Debugger records the levels between 6 and 9.</p> <p>For situations where the stack is very large, it is recommended that you</p>	Recorder Toolbar ^[2163] Breakpoint & Markers ^[2157] Control the Recording Session ^[2161]

Topic	Detail	See also
	first use a low stack depth of 2 or 3 and gradually increase it if necessary to expand the picture.	

15.2.3 Place Recording Markers

This section explains how to place record markers, which enable you to silently record code execution between two points. The recording can be used to generate a Sequence diagram.

As this process records the execution of multiple threads, it can be particularly useful in capturing event driven sequences (such as mouse and timer events).

Access [Analyzer | Breakpoints & Events](#)

Use to

- Set recording markers, to define what is to be recorded during code execution in order to generate a Sequence diagram

Topics

Topic	Link
Different recording markers can be used for recording the execution flow; see the related links for information on the properties and usage of these markers.	Marker Types ^[2155] Set Record Markers ^[2155]
How to manage breakpoints in the Breakpoint & Markers window.	The Breakpoint & Events Window ^[2157]
How to activate and deactivate markers.	Breakpoint and Marker Management ^[1863]
Working with Marker Sets - when you create a breakpoint or marker, it is automatically added to a marker set, either the Default set or a set that you create for a specific purpose.	Working with Marker Sets ^[2158] Recording Activity for a Class ^[2160]

Notes

- The *Breakpoint and Marker Management* topic (Software Engineering) covers a different perspective

Learning Center topics

- **(Alt+F1) | Execution Analysis | Recording Execution**

15.2.3.1 Set Record Markers

Markers are set in the source code editor. Like breakpoints, they are placed on a line of code; when that line of code executes, the Execution Analyzer performs the code execution recording action appropriate to the marker.

Access Click on the Class containing code: **F12**

How to

To set a recording marker

Step	Action	See also
1	Press F12 to open the source code to debug, in the integrated source code editor.	Editing Source Code ^[1799]
2	Find the appropriate code line and right click in the left (Breakpoint) margin to bring up the breakpoint/marker context menu; select the required marker type: <div data-bbox="367 974 1133 1400" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> <pre> 12 Passengers = GetRandom(0,200); 13 Number = number; 14 Network = Net... </pre> </div>	Marker Types ^[2155]
3	If a Start Recording Marker has been set, you must also set an End Recording Marker .	

15.2.3.2 Marker Types

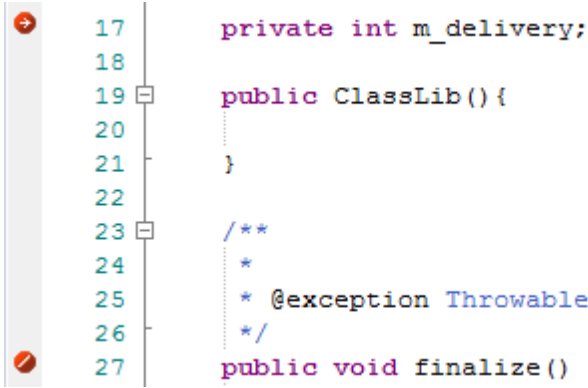
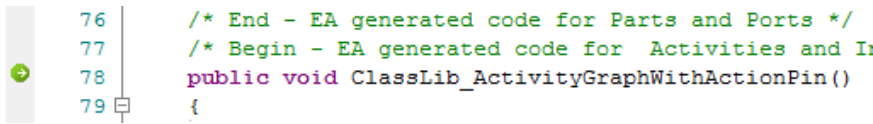
Recording markers are similar to breakpoints; however, instead of breaking execution as it does at a breakpoint, the debugger takes the action defined by the type of marker. If the marker is denoted as a recording *start point*, the debugger immediately begins to trace all executed calls from that point for the breaking thread. Recording is stopped again when either the thread that is being captured terminates or the thread encounters a recording *end point*.

Recording markers are set in the source code editor. If you right-click on the breakpoint margin at the point to begin recording, a context menu displays that enables you to select the type of a marker.

Use to

- Set Start and End points for recording
- Set Markers for a Single Method
- Set Stack Trace Markers

Reference

Marker	Detail	See also
<p>Start Recording marker</p>	<p>Select the Add Start Recording Marker option, then right-click on the breakpoint margin at the point to stop recording and select the Add End Recording Marker context menu option; the markers are shown below:</p>  <p>When the debugger is executed it continues to run the thread, recording a stack history until either the End Recording marker is encountered or the thread terminates, unlike normal breakpoints where the debugger halts and displays the line of code.</p> <p>In order to limit the amount of information being collected, you can control the stack depth being recorded.</p>	<p>The Breakpoints & Events Window^[2157]</p> <p>Analyzer Script Editor^[1837]</p> <p>Control Stack Depth^[2153]</p>
<p>Stack Auto-Capture marker</p>	 <p>(Native Code only) Stack markers enable you to capture any unique stack traces that occur at a point in an application; they provide a quick and useful picture of where a point in an application is being called from.</p> <p>To insert a marker at the required point in code, right-click on the line and select the Add Stack Auto Capture Marker context menu option.</p> <p>Each time the debugger encounters the marker it performs a stack trace; if the stack trace is not in the recording history, it is copied, and the application continues running.</p>	
<p>Method Auto Record marker</p>	<p>A Method Auto Record marker enables you to record activity for a particular function during a debug session; the debugger records any function calls executed after the marker point, and always stops recording when this function exits.</p>	

Marker	Detail	See also
	<p>The function marker combines a Start Recording marker and an End Recording marker in one.</p> <pre> 185 // 186 // CRecurrenceDlg message handlers 187 188 BOOL CRecurrenceDlg::OnInitDialog() 189 { 190 CBCGPDIALOG::OnInitDialog(); 191 192 UINT nMask = 193 CBCGPDIALOG::DTM_SPIN 194 CBCGPDIALOG::DTM_DATE 195 CBCGPDIALOG::DTM_TIME 196 CBCGPDIALOG::DTM_CHECKBOX 197 CBCGPDIALOG::DTM_DROPCALENDAR 198 CBCGPDIALOG::DTM_CHECKED; 199 200 UINT nFlags = CBCGPDIALOG::DTM_CHECKED CBCGPDIALOG::DTI 201 //----- 202 // Setup date fields: </pre>	

Learn more

- [Set Record Markers](#) ^[2155]
- [Recording Activity for a Class](#) ^[2160]

Learning Center topics

- (Alt+F1) | [Execution Analysis | Recording Execution | Method Record Marker](#)

15.2.3.3 The Breakpoints & Events Window

The Breakpoints & Events window enables you to apply control to Visual Execution Analysis when recording execution to generate Sequence diagrams.

The window also contains tabs for managing Simulation Markers and Simulation Events.

Access [Analyzer | Breakpoints & Events](#)

Topics

Topic	Detail	See also
Usage	<p>Use the Breakpoints & Events window to enable, disable and delete markers, and manage them as sets.</p> <p>You can organize how the markers are displayed, either in list view or grouped by file or Class.</p>	<p>Breakpoint and Marker Management ^[1869]</p> <p>Working with Marker Sets ^[2158]</p>

Learn more

- [Simulation Breakpoints](#) ^[2114]
- [Simulation Events](#) ^[2125]

Learning Center topics

- (Alt+F1) | **Execution Analysis | Recording Execution**


15.2.3.4 Working with Marker Sets

Marker sets enable you to create markers as a named group, which you can reapply to a code file for specific purposes.

You can perform certain operations from the Breakpoints & Events window alone, but to understand and use markers and marker sets you should also display the appropriate code file in the Source Code Viewer (click on the Class element and press **(F12)**).

Access **Analyzer | Breakpoints & Events: Set Options toolbar icon**

Topics

Topic	Detail	See also
Example of Use	<p>You might create a set of Method Auto Record markers to record the action of various functions in the code, and a set of Stack Capture markers to record the sequence of calls that cause those functions to be called.</p> <p>You could then create Sequence diagrams from the recordings under each set.</p>	Marker Types ^[2155]
Create a Marker Set	<p>To create a marker set from the Breakpoints & Events window, click on the drop-down arrow on the Set Options icon () and select the New Set option.</p> <p>The New Breakpoint Marker Set dialog displays; in the Enter New Set Name field, type a name for the set, and click on the Save button.</p> <p>The set name displays in the text field to the left of the <i>Set Options</i> icon.</p> <p><i>Alternatively</i>, you can either:</p> <ul style="list-style-type: none"> • Create a Class Activity marker set or • Select the Save as Set option from the Set Options drop-down, to make an exact copy of the currently-selected set, which you can then edit 	Recording Activity for a Class ^[2160]
Accessing Sets	<p>To access a marker set, click on the drop-down arrow on the text field to the left of the Set Options icon, and select the required set from the list.</p> <p>The markers in the set are listed in the Breakpoints & Events window.</p>	

Topic	Detail	See also
	<p>You would normally load a marker set prior to the point at which an action is to be captured.</p> <p>For example, to record a sequence involving a particular dialog, when you begin debugging you would load the set prior to invoking the dialog; once you bring up the dialog in the application, the operations you have marked are recorded.</p>	
Add Markers to Set	<p>To add markers to a marker set, add each required marker to the appropriate line of code in the Source Code Viewer.</p> <p>The marker is immediately added to whichever set is currently listed in the Breakpoints & Events window.</p> <p>Each marker listed on the dialog has a checkbox in the Enabled column; newly-added markers are automatically enabled, but you can disable and re-enable the markers quickly as you check the code.</p>	Set Record Markers ^[2159]
Storage of Sets	<p>When you create a marker set it is immediately saved within the model; any user using the model has access to that set.</p> <p>However, the Default set, which always exists for a model, is a personal workspace, is not shared and is stored external to the model.</p>	
Delete a Marker from a Set	<p>Right-click on the marker and select the Delete Breakpoint context menu option.</p>	
Delete a Set	<p>If you no longer require a marker set, access it on the Breakpoints & Events window and select the Delete Selected Set option from the Set Options drop-down list.</p> <p>You can also clear <i>all</i> user-defined marker sets by selecting the Delete all sets option; a prompt displays to confirm the deletion.</p>	

Notes

- Marker Sets are very simple and flexible but, as they are available for use by any user of the model, they can be easily corrupted; consider the following guidelines:
 - When naming a set, use your initials in the name and try to indicate its use, so that other model users can recognize its owner and purpose
 - When using a set other than **Default**, avoid excessive experimentation so that you don't add lots of ad-hoc markers to the set
 - Make sure you are aware of which marker set is exposed in the Breakpoints & Events window, as you can easily inadvertently add markers to the set that are not relevant to the code file the set was created for
 - In any set, if you have added markers that don't have to be kept, delete them to maintain the purpose of the set; this is especially true of the Default set, which can quickly accumulate

redundant ad-hoc markers

Learning Center topics

- (Alt+F1) | [Execution Analysis | Recording Execution | Creating a Marker Set](#)
- (Alt+F1) | [Execution Analysis | Recording Execution | Enable a Marker Set](#)

15.2.3.5 Recording Activity for a Class

In addition to setting breakpoints and markers in the code editor, or creating a *marker set* through the Breakpoints & Events window, you can record all the operations of a Class or a subset using the Class Markup Selection dialog to create a marker set of record markers for those operations. These marker sets are also available to all users of the model.

Access [Project Browser Class context menu | Execution Analyzer | Markup Class For Recording](#)

Reference

Field	Usage	See also
Existing marker set	Select to change the operations, marker types or stack depth of an existing marker set.	
New marker set	Select to create a new marker set.	
Name	Type the name of the marker set to create or edit.	
Operation check boxes	All checkboxes default to selected; click on the checkbox against each operation that you <i>do not</i> want to record. Click again on the checkbox against any excluded operation that you do now want to record, to re-select it.	
Include disabled operations	Select this option to set a recording marker against all operations, but disable all those that you have not specifically selected above. In the Breakpoints & Events window, you can then enable these markers individually or as a whole, rather than enabling them by editing the marker set.	Working with Marker Sets <small>[2158]</small>
Marker Type	Click on the drop-down arrow and select the type of marker to set against each of the selected operations. The marker type specifies the action to take when the process encounters that marker on each operation: <ul style="list-style-type: none"> • Record function • Record stack trace • Break execution 	Marker Types <small>[2158]</small>

Field	Usage	See also
Limit recording frame depth	Set the stack depth to limit the recording, to avoid ultimately producing Sequence diagrams that are too complicated to read.	Control Stack Depth ^[2153]
OK	Click to store the marker set under the name you have specified; the set can then be loaded either before or during a session, from the Breakpoint & Markers window toolbar.	Breakpoint and Marker Management ^[1869]

Learning Center topics

- (Alt+F1) | [Execution Analysis](#) | [Recording Execution](#) | [Automated Set Creation](#)

15.2.4 Control the Recording Session

The Record & Analyze window enables you to control a recording session. The control has a toolbar, and a history window that displays the recording history as it is captured. Each entry in this window represents a call sequence made up of one or more function calls.

Access [Analyzer](#) | [Record](#) | [Recorder](#)

You must also open the Execution Analyzer window ([Analyzer](#) | [Execution Analyzer](#)), which lists all the scripts in the model; make sure you select and activate the appropriate script for the recording.

Learn more

- [Recorder Toolbar](#) ^[2163]
- [Start Recording](#) ^[2161]
- [The Recording History](#) ^[2149]

15.2.4.1 Start Recording

You start a recording by selecting the **Recording** icon on the Record & Analyse window toolbar. However, before you start recording, select the package or Class under which to generate the Sequence diagram, in the Project Browser.

The Record dialog displays, on which you can optionally select:

- A marker set to use
- A State Transition diagram to record during the session; the Execution Analyzer captures instances of the Class that the State Machine refers to and calculates the State whenever a function in the recorded stack exits

Field/Button	Detail	See also
Using record set:	Recording markers determine what is recorded. If you have a recording set to use, click on the drop-down arrow and select it.	Working with Marker Sets ^[2158]


Field/Button	Detail	See also
Using Statechart / Diagram:	<p>The recording can also capture changes in State transitions.</p> <p>If you have a State Machine diagram to use you can browse for and select the diagram, using the State Transition Recorder dialog.</p>	<p>Reporting State Transitions ^[2165]</p> <p>Reporting a State Machine ^[2166]</p>
OK	<p>Click on this button to invoke the start screen of the application to be recorded.</p> <p>The application screen should have a Start or Run button; click on this to execute the application.</p> <p>When the application has run for an appropriate period of number of cycles, click on the Stop or Cancel button.</p> <p>The execution is recorded and you can generate the Sequence diagram from this record.</p>	<p>Generating Sequence Diagrams ^[2164]</p>

Notes

- The State Machine diagram must exist as a child of a Class, and can exist anywhere in the model; wherever possible, it should be created as a child of the Class to which it relates

15.2.4.2 Step Through Function Calls





Topics







Topic	Detail	See also
Stepping Through	<p>The Step Through function can be executed by clicking on the Step Through button on the Record & Analyze window toolbar.</p>  <p>Alternatively, press (Shift+F6) or select the Analyzer Debug Step Into context menu option.</p> <p>The <i>Step Through</i> command causes a <i>Step Into</i> command to be executed; if any function is detected, then that function call is recorded in the History window.</p> <p>The debugger then steps out, and the process can be repeated.</p> <p>This button enables you to record a call without having to actually step into a function; the button is only enabled when at a breakpoint and in manual recording mode.</p>	

15.2.4.3 Recorder Toolbar

The Record & Analyze window toolbar provides the facilities for starting, stopping and moderating an execution analysis recording session.

Reference

Icon	Description	See also
	<p>The Recording icon is enabled when the active Analyzer Script is configured for debugging.</p> <p>Use this icon to launch and record the application described in the script.</p> <p>You can optionally select a record marker set and / or a State machine to use during the recording.</p>	<p>Start Recording ^[2161]</p> <p>Recording Activity for a Class ^[2160]</p>
	<p>The Recording (attach to process) icon is enabled even if no Analyzer Script exists.</p> <p>When you use this icon, a prompt displays to select a process to record and a debugging platform to use.</p> <p>You can also optionally select a record marker set and / or a State machine to use during the recording.</p>	<p>Start Recording ^[2161]</p> <p>Recording Activity for a Class ^[2160]</p>
	<p>Use the Manual Record icon to perform ad-hoc recording of the current thread during a debug session.</p> <p>Used in conjunction with the 'step' buttons of the debugger; each function that is called due to a step command is logged to the history window.</p> <p>This icon is only enabled if no recording is taking place and you are currently at a breakpoint (that is, debugging).</p>	
	<p>Use the Auto Record Selected Thread icon to perform ad-hoc auto-recording during a debug session.</p> <p>When you select this icon, the Analyzer begins recording and does not stop until either the program ends, you stop the debugger or you click on the Stop icon.</p> <p>This icon is only enabled if no recording is taking place and you are currently at a breakpoint (that is, debugging).</p>	
	<p>Use the Step Through icon to step into a function, record the function call in the History window, and step back out.</p> <p>Enabled for manual recording only.</p>	<p>Step Through Function Calls ^[2162]</p>
	<p>Use the Stop Recording icon to stop recording the execution trace.</p>	

Icon	Description	See also
	Use this field to define the marker set's recording stack depth - the number of frames from the point recording began.	Marker Types ^[2155]
	Select the Show/Hide lines during recording icon to show live line-by-line execution in the code editors as the program runs. This icon is disabled if marker sets are in use.	
	Use this icon to select an XML file from which to restore a previously-saved recording history.	
	Use the Save Recording History to file icon to save the recording history to an XML file.	
	Use the Generate Sequence Diagram from recording icon to generate a Sequence/State diagram from the Execution Analyzer trace.	Generating Sequence Diagrams ^[2164]
	Use the Testpoints icon to generate a TestPoints diagram from the Execution Analyzer trace, including Classes, Test Cuts and a Test Set that can be used with the Testpoint facility.	Testpoint Management ^[2187] Combine Testpoints ^[2196]

15.2.5 Generating Sequence Diagrams

This topic describes what you might do with the recording of an execution analysis session.





Access [Analyzer](#) | [Record](#) | [Recorder](#)

Use to

- Generate a Sequence diagram from a recorded execution analysis session, for:
 - all recorded sequences or
 - a single sequence in the session
- Save the recorded sequence to file
- Retrieve the saved recording and load it into the Record & Analyze window

Reference

Action	Detail	See also
Generate a diagram	Select the appropriate package in the Project Browser, in which to store the Sequence diagram.	The Recording History ^[2149]

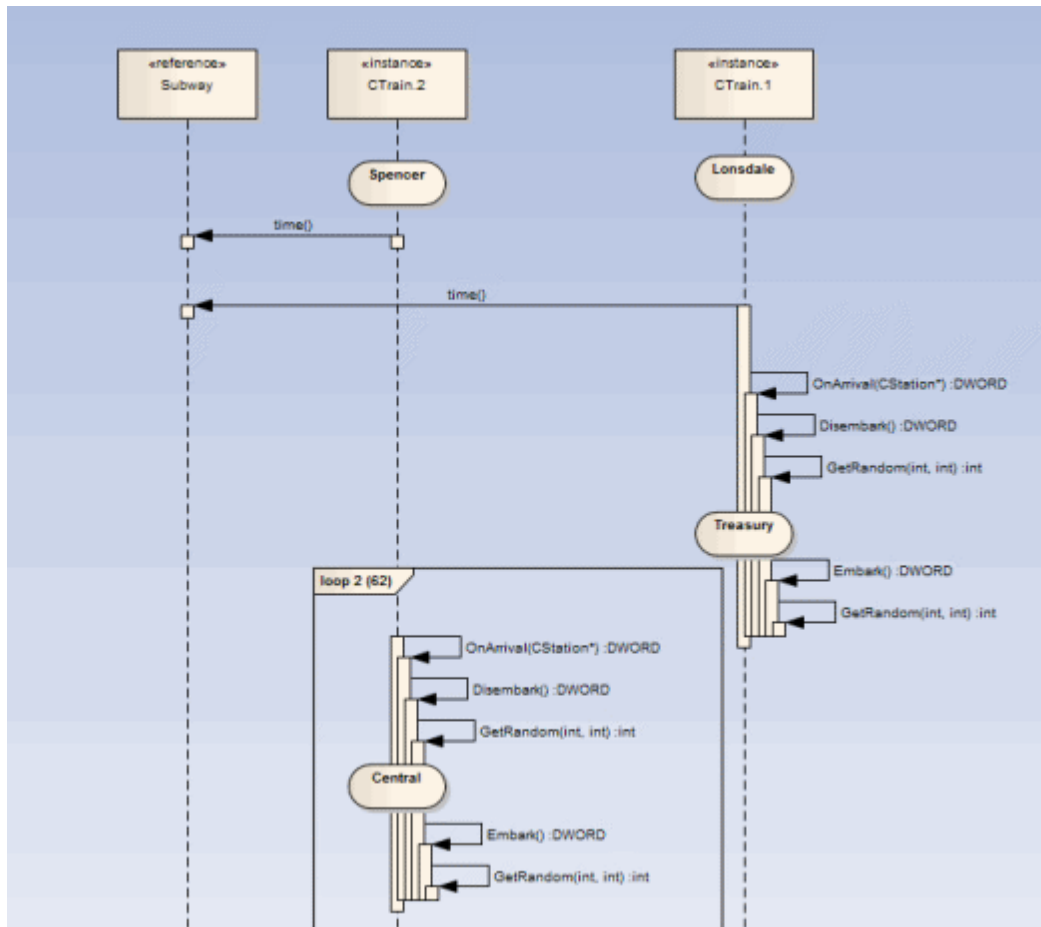
Action	Detail	See also
	<p>To create the diagram from all recorded sequences, either:</p> <ul style="list-style-type: none"> Click on the Create Sequence Diagram icon () in the Record & Analyze window toolbar, or Right-click on the body of the window and select the Generate Sequence Diagram context menu option <p>To create the diagram from a single sequence, either:</p> <ul style="list-style-type: none"> Select the sequence and click on the Create Sequence Diagram icon () in the Record & Analyze window toolbar, or Right-click on the sequence and select the Generate Diagram from Selected Sequence context menu option 	Recorder Toolbar ^[2163] Working With Recording History ^[2151]
Save a recorded sequence to an XML file	<p>Click on the sequence and click on the Record & Analyze window toolbar Save icon ()</p>	Recorder Toolbar ^[2163]
Access an existing sequence XML file	<p>Either:</p> <ul style="list-style-type: none"> Click on the toolbar Open icon () , or Right-click on a blank area of the screen and click on the Load Sequence From File context menu option <p>The Windows Open dialog displays, from which you select the file to open.</p>	

15.2.6 Reporting State Transitions

This section describes how you can generate Sequence diagrams that show transitions in state as a program executes.

Use to

- Generate Sequence diagrams that report user-defined transitions in state as a program executes (as shown in the example generated diagram below)

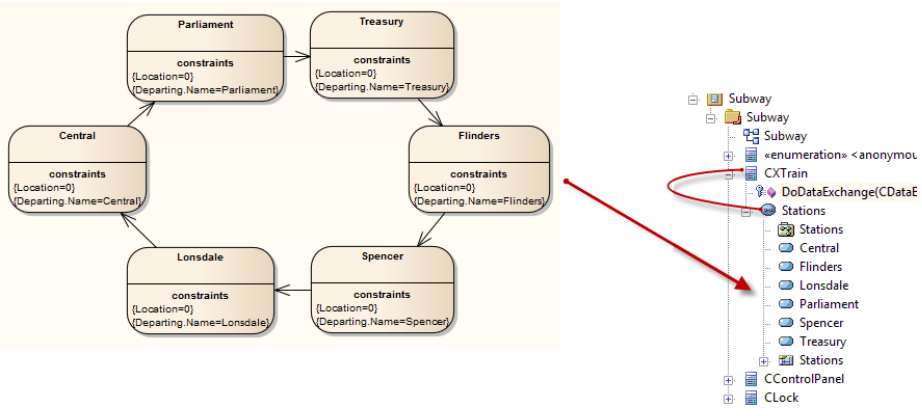


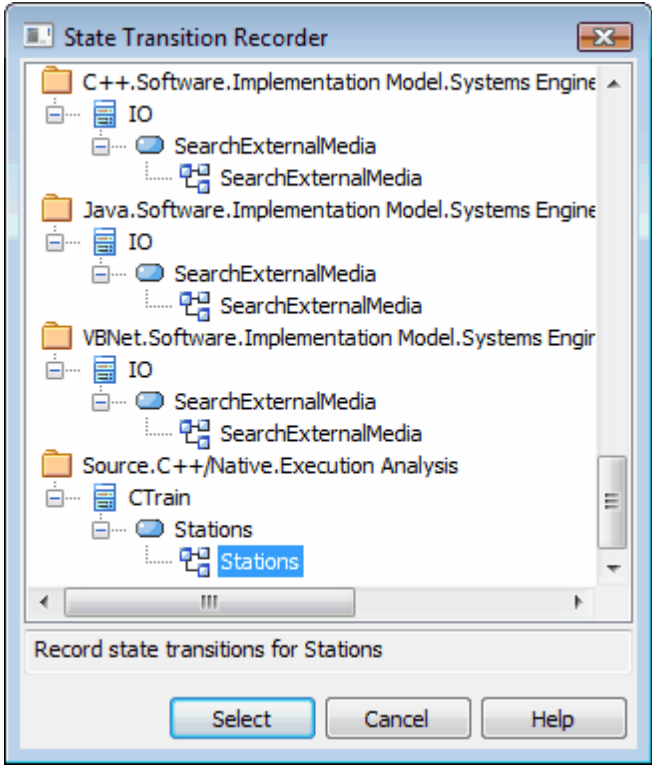
Topic	Link
Create a State Machine under the Class to be reported.	Reporting a State Machine ^[2166]
Set the constraints against each State to define the change in state to be reported.	Recording and Mapping State Changes ^[2168]

15.2.6.1 Reporting a State Machine

A State Machine can be used to illustrate how States change during the execution of an application, explicitly describing the transitions from State to State. You can show these State transitions on the Sequence diagrams you generate in Execution Analysis.

Topic	Detail	See also
Process	<p>Firstly you create a State Machine to model all the valid system states, as a child of the appropriate Class element, and on the State Machine you create the State elements that correspond to any states to be captured for your Class.</p> <p>You then set constraints in the States to record and map the State changes.</p> <p>When you record code execution for the Class, you set an option in the Record dialog and browse for the State Machine using the State Transition dialog.</p> <p>The State Transition dialog presents a list of State Machines for the entire model, in</p>	<p>State Machines ^[1032]</p> <p>Recording and Mapping State Changes ^[2168]</p> <p>Start</p>



Topic	Detail	See also
	<p>which you locate and select the appropriate diagram (see the Example below).</p> <p>When you generate the Sequence diagram, it depicts not only the sequence but changes in State at the various points in the sequence; each Class instance participating in the detection process is displayed with its own lifeline.</p>	<p>Recording 2161</p> <p>Reporting State Transitions 2165</p>
<p>Example</p>	<p>The State Machine below, called <i>Stations</i>, shows the different States within the Melbourne Underground Loop subway system.</p> <p>A train traveling on the subway network can be stopped at any of the stations represented on the State Machine.</p> <p>The <i>Stations</i> State Machine is a child of the <i>CTrain</i> Class.</p>  <p>When you browse for the diagram in the State Transition Recorder dialog, the hierarchy shows only the root package, parent Class and child Sub Machine and diagram; no other model components are listed.</p>	

Topic	Detail	See also
		

15.2.6.2 Recording and Mapping State Changes

This topic discusses how to set constraints against each State in the State Machine under a Class, to define the change in state to be recorded.

Topics

Topic	Detail	See also									
Example	<p>The State Properties dialog shown below is for the State called <i>Parliament</i>; the Constraints tab is open to show how the State is linked to the Class <i>CXTrain</i>.</p> <p>A State can be defined by a single constraint or by many; in the example below the State <i>Parliament</i> has two constraints:</p> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> <p>Defined Constraints   New Save Delete</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 60%;">Constraint</th> <th style="width: 20%;">Type</th> <th style="width: 20%;">Status</th> </tr> </thead> <tbody> <tr> <td>Location=0</td> <td>Invariant</td> <td>Approved</td> </tr> <tr> <td>Departing.Name=Parliament</td> <td>Invariant</td> <td>Approved</td> </tr> </tbody> </table> </div> <p>The values of constraints can only be compared for <i>elemental</i>, <i>enum</i> and <i>string</i> types</p> <p>The <i>CXTrain</i> Class has a member called <i>Location</i> of type <i>int</i>, and a</p>	Constraint	Type	Status	Location=0	Invariant	Approved	Departing.Name=Parliament	Invariant	Approved	<p>Reporting a State Machine [2166]</p>
Constraint	Type	Status									
Location=0	Invariant	Approved									
Departing.Name=Parliament	Invariant	Approved									

Topic	Detail	See also
	<p>member called <i>Departing.Name</i> of type <i>CString</i>; what this constraint means is that this State is evaluated to true when:</p> <ul style="list-style-type: none"> • an instance of the <i>CXTrain</i> Class exists and • its member variable <i>Location</i> has the value 0 and • the member variable <i>Departing.Name</i> has the value Parliament 	
Operators in Constraints	<p>There are two types of operators you can use on constraints to define a State:</p> <ul style="list-style-type: none"> • Logical operators AND and OR can be used to combine constraints • Equivalence operators {= and !=} can be used to define the conditions of a constraint <p>All the constraints for a State are subject to an AND operation unless otherwise specified; you can use the OR operation on them instead, so you could rewrite the constraints in the above example as:</p> <pre> Locat i on=0 OR Locat i on=1 AND Depar t i ng. Name! =Cent r al </pre> <p>Below are some examples of using the equivalence operators:</p> <pre> Depar t i ng. Name! =Cent r al AND Locat i on! =1 </pre>	

Notes

- Quotes around strings are optional; the comparison for strings is always case-sensitive in determining the truth of a constraint

Learn more

- [Reporting a State Machine](#) 

15.3 Profiling Native Applications

Access [Analyzer](#) | [Profile](#)

Topics

Topic	Detail	See also																																
Visual Execution Profiler	<p>The Visual Execution Profiler enables you to quickly report on:</p> <ul style="list-style-type: none"> • The most frequently called functions in a running application • Tasks in an application that are taking more time than expected • Which functions are taking the most time in an application <p>The Profiler only works with MS Native Windows applications, but can be used under WINE (Linux and Mac) to debug standard Windows applications deployed in a WINE environment.</p> <p>The Profiler can generate a report that shows how these functions are called in relation to the application; the following image shows how one function is taking most of the overall time recorded, and how that is broken up into some of the functions that it calls.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #e0e0e0;">Stack</th> <th style="background-color: #e0e0e0;">Inclusive Hits</th> <th style="background-color: #e0e0e0;">Hits</th> <th style="background-color: #e0e0e0;">Inclusive Hi...</th> </tr> </thead> <tbody> <tr> <td>[-] CBCGPVisualManager2010::OnUpdateSystemColors</td> <td style="text-align: center;">200</td> <td></td> <td style="text-align: center;">72%</td> </tr> <tr> <td>[-] CBCGPTagManager::ExcludeTag</td> <td style="text-align: center;">85</td> <td></td> <td style="text-align: center;">31%</td> </tr> <tr> <td>[-] mfc90ud</td> <td style="text-align: center;">84</td> <td></td> <td style="text-align: center;">30%</td> </tr> <tr> <td>[-] BCGCBPRO1100ud90</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0%</td> </tr> <tr> <td>[-] CBCGPControlRenderer::Create</td> <td style="text-align: center;">32</td> <td></td> <td style="text-align: center;">12%</td> </tr> <tr> <td>[-] CBCGPTagManager::ReadControlRenderer</td> <td style="text-align: center;">62</td> <td></td> <td style="text-align: center;">22%</td> </tr> <tr> <td>[-] CBCGPTagManager::ParseControlRenderer</td> <td style="text-align: center;">56</td> <td></td> <td style="text-align: center;">20%</td> </tr> </tbody> </table>	Stack	Inclusive Hits	Hits	Inclusive Hi...	[-] CBCGPVisualManager2010::OnUpdateSystemColors	200		72%	[-] CBCGPTagManager::ExcludeTag	85		31%	[-] mfc90ud	84		30%	[-] BCGCBPRO1100ud90	1	1	0%	[-] CBCGPControlRenderer::Create	32		12%	[-] CBCGPTagManager::ReadControlRenderer	62		22%	[-] CBCGPTagManager::ParseControlRenderer	56		20%	
Stack	Inclusive Hits	Hits	Inclusive Hi...																															
[-] CBCGPVisualManager2010::OnUpdateSystemColors	200		72%																															
[-] CBCGPTagManager::ExcludeTag	85		31%																															
[-] mfc90ud	84		30%																															
[-] BCGCBPRO1100ud90	1	1	0%																															
[-] CBCGPControlRenderer::Create	32		12%																															
[-] CBCGPTagManager::ReadControlRenderer	62		22%																															
[-] CBCGPTagManager::ParseControlRenderer	56		20%																															

Notes

- The Profiler, or sampler, is available in the Enterprise Architect Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn more

- [System Requirements](#) ^[2171]
- [Getting Started](#) ^[2171]
- [Profiler Operation](#) ^[2174]

15.3.1 System Requirements

Profiles can monitor calls in an application that is configured to be debugged in the Execution Analyzer.

Options:

Profiles can be recorded by:

- Executing an existing Analysis script for an application, or
- Attaching to a running process

The latter option is useful if the development application being analysed:

- Spawns from a call from another application (such as Winword.exe)
- Is running as a Service
- Is best monitored at a known point of processing (e.g. after a user interaction)

Topics

Topic	Detail	See also
Prerequisites	<p>The Profiler requires an Analyzer Build & Debug script for a C, C++ or Visual Basic application that has been compiled with the Microsoft™ Native compiler; note that this requires a PDB file to be available.</p> <p>The Profiler can sample both debug and release configurations of an application, provided that the PDB file for each executable exists and is up to date.</p>	Debugging ^[1867] Microsoft C++ and Native (C, VB) ^[1841] General Setup ^[1841]

Learn more

- [Getting Started](#)^[2171]
- [Profiling Native Applications](#)^[2170]

Learning Center topics

- **(Alt+F1) | Execution Analysis | Profiling Native Code**

15.3.2 Getting Started








Access **Analyzer | Profile**



Topics

Topic	Detail	See also
Abstract	The Profiler operates by taking samples of a process at intervals of up to 250 milliseconds.	

Topic	Detail	See also
	<p>At these intervals the Profiler interrupts the process and collects stack information for all threads running at that time; this information is sent back to Enterprise Architect where it is collected sorted and stored.</p> <p>You can pause and resume profiling at any time during the session, or clear any sample data collected and begin again.</p> <p>If you stop the Profiler and the process is still running, you can quickly attach to it again.</p>	






Toolbar Options:

Icon	Action	See also
	(When an application is configured for the package) create the Profiler process, which launches the configured application.	Managing Scripts <small>[1828]</small>
	Profile an application that is already running.	
	<p>When the application is running, pause and resume sample capture.</p> <p>Pausing sampling enables the Report and Erase buttons.</p>	
	Stop the Profiler process; if any samples have been collected, the Report button is enabled.	
	Generate a report on the current number of samples collected.	Profiling Native Application <small>[2170]</small>
	Set the interval, in milliseconds, at which samples are taken of the target process; the range of possible values is 1 - 250 .	Setting Options <small>[2174]</small>
	<p>Set Profiler options, using a drop-down menu; the options are:</p> <ul style="list-style-type: none"> • Load Report from Disk - load and display a previously-generated report from an XML disk file • Analyzer Scripts (Shift+F12) - display the Execution Analyzer window to enable creation or editing of scripts and debug configuration • Start Sampling Immediately - begin sample collection immediately upon either process start (main thread entry point executed) or attachment of process by Profiler 	Analyzer Scripts <small>[1827]</small>

Icon	Action	See also
	<ul style="list-style-type: none"> • Capture Debug output - capture any appropriate debug output and redirect it to the Enterprise Architect Output window • Stop Process on Exit - select to terminate the target process when the Profiler is stopped 	
	Erase the collected data.	
	Display the Help topic for this window.	

15.3.3 Start & Stop the Profiler

Topics

Topic	Detail	See also
Usage	<p>For most debugging operations it is necessary to have first configured an Execution Analyzer Script that typically defines the application to build, test and debug, and any sequence recording options.</p> <p>It is possible to use the Profiler without doing any of this by using the Attach to Process button.</p> <p>If the application to profile is the one defined in the current package, use the Launch button.</p> 	
Buttons	 (When an application is configured for the package) create the Profiler process, which launches the configured application.	Managing Scripts 
	 Profile an application that is already running.	
	 Stop the Profiler process.	

15.3.4 Profiler Operation

Topics

Topic	Detail	See also
Usage	<p>Enterprise Architect creates a Profiler process whenever you click on the Launch or Attach to Process button on the Profiler window toolbar; this process operates by collecting samples from the stacks of every thread in the target process.</p> <div data-bbox="349 633 1195 817" data-label="Diagram"> <pre> classDiagram class EA class Profiler class TargetApplication TargetApplication --> Profiler : «flow» Profiler --> EA : «flow» </pre> </div> <p>The sampler process exits if you click on the Stop button, if the target application terminates, or if you close the current model.</p> <p>You can turn sample collection on and off at any time during a session.</p> <p>When sampling is turned on or resumed, the Profiler process becomes active and samples are collected from the target; resuming sampling collects completely new samples.</p> <p>The Profiler process idles if sampling is turned off or paused during a session. The Report and Erase buttons then become enabled.</p> <p>Click on the Report button to produce a call graph summary similar to that in the <i>Profile Native Applications</i> topic; this report can be saved to file.</p> <p>Click on the Erase button to discard any samples currently collected for the target.</p>	<p>Profile Native Applications ²¹⁷⁰</p>

15.3.5 Setting Options


Topics

Topic	Icon	Detail	See also
Interval		Set the interval, in milliseconds, at which samples are taken of the target process; the range of possible values is 1 - 250 .	
Profile Options		<p>Set Profiler options, using a drop-down menu; the options are:</p> <ul style="list-style-type: none"> • Start Sampling Immediately - begin sample collection immediately upon either process start (main thread entry point executed) or attachment of process by Profiler • Capture Debug output - capture any appropriate debug output and redirect it to the Enterprise Architect Output window 	

Topic	Icon	Detail	See also
		<ul style="list-style-type: none"> • Stop Process on Exit - select to terminate the target process when the Profiler is stopped 	

15.3.6 Save and Load Reports

Topics

Topic	Detail	See also																																
Usage	<p>The Profiler Reports can be saved in either binary format or xml format.; save the report using the toolbar above the report (Stack) view.</p> <table border="1"> <thead> <tr> <th>Stack</th> <th>Inclusive Hits</th> <th>Hits</th> <th>Inclusive Hi...</th> </tr> </thead> <tbody> <tr> <td>[-] CBCGPVisualManager2010::OnUpdateSystemColors</td> <td>200</td> <td></td> <td>72%</td> </tr> <tr> <td>[-] CBCGPAdapterManager::ExcludeTag</td> <td>85</td> <td></td> <td>31%</td> </tr> <tr> <td>[-] mfc90ud</td> <td>84</td> <td></td> <td>30%</td> </tr> <tr> <td>[-] BCGCBPRO1100ud90</td> <td>1</td> <td>1</td> <td>0%</td> </tr> <tr> <td>[-] CBCGPControlRenderer::Create</td> <td>32</td> <td></td> <td>12%</td> </tr> <tr> <td>[-] CBCGPAdapterManager::ReadControlRenderere</td> <td>62</td> <td></td> <td>22%</td> </tr> <tr> <td>[-] CBCGPAdapterManager::ParseControlRenderere</td> <td>56</td> <td></td> <td>20%</td> </tr> </tbody> </table> <p>To load a report, use the Profiler Toolbar Options button  and select the Load Report From Disk option.</p>	Stack	Inclusive Hits	Hits	Inclusive Hi...	[-] CBCGPVisualManager2010::OnUpdateSystemColors	200		72%	[-] CBCGPAdapterManager::ExcludeTag	85		31%	[-] mfc90ud	84		30%	[-] BCGCBPRO1100ud90	1	1	0%	[-] CBCGPControlRenderer::Create	32		12%	[-] CBCGPAdapterManager::ReadControlRenderere	62		22%	[-] CBCGPAdapterManager::ParseControlRenderere	56		20%	
Stack	Inclusive Hits	Hits	Inclusive Hi...																															
[-] CBCGPVisualManager2010::OnUpdateSystemColors	200		72%																															
[-] CBCGPAdapterManager::ExcludeTag	85		31%																															
[-] mfc90ud	84		30%																															
[-] BCGCBPRO1100ud90	1	1	0%																															
[-] CBCGPControlRenderer::Create	32		12%																															
[-] CBCGPAdapterManager::ReadControlRenderere	62		22%																															
[-] CBCGPAdapterManager::ParseControlRenderere	56		20%																															

15.3.7 Save Report in Team Review

You can save any current report as a resource for a Category, Topic or Post in the Team Review. The report can then be shared and reviewed at any time as it is saved with the model.

Access [Team Review Context Menu](#) | [Share Resource](#) | [Add Active Profiler Report](#)

Use to

- Preserve a profiler report to compare against future runs
- Allow other people to investigate the profile

Learn more

- [\(Team Review\) Context Menu](#) ²⁸⁰

15.4 Object Workbench

This section describes the Object Workbench, a tool in Enterprise Architect debugging that enables you to create your own Class workbench instances and invoke methods on them.

Topic	Link
Setup requirements for the workbench.	Workbench Setup ^[1861]
Overview of the workbench as a debugger tool.	How it Works ^[2176]
How to create and delete workbench instances.	Create & Delete Workbench Instances ^[2177]
How to invoke methods to record the Stack trace and produce Sequence diagrams.	Invoke Methods ^[2179]

Learning Center topics

- (Alt+F1) | **Execution Analysis | Object Workbench**

15.4.1 How it Works

The Workbench is a tool in Enterprise Architect Debugging, enabling you to create your own variables and invoke methods on them. Stack trace can be recorded and Sequence diagrams produced from the invocation of such methods. It provides a quick and simple way to debug your code.

Topics

Topic	Detail	See also
Mode	<p>The Workbench operates in two modes:</p> <ul style="list-style-type: none"> • Idle mode - When the Workbench is in idle mode, instances can be created and viewed and their members inspected • Active mode - When methods are invoked on an instance, the Workbench enters Active mode and the displayed instances change if the debugger encounters any breakpoints; if no breakpoints are set, then the instances do not change and the Workbench immediately returns to Idle mode 	
Logging	The results of creating instances and the results of calls on their methods are displayed in the Debug window.	Run the Debugger ^[1875]

Learn more

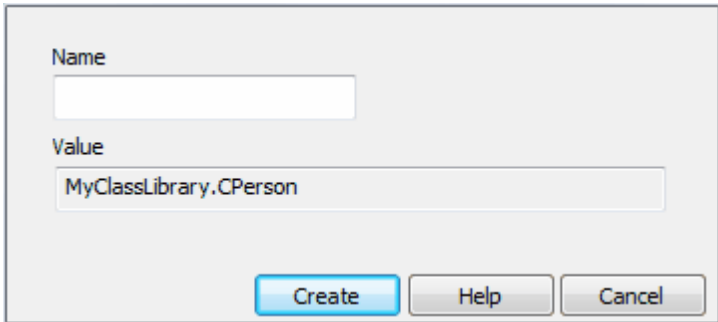
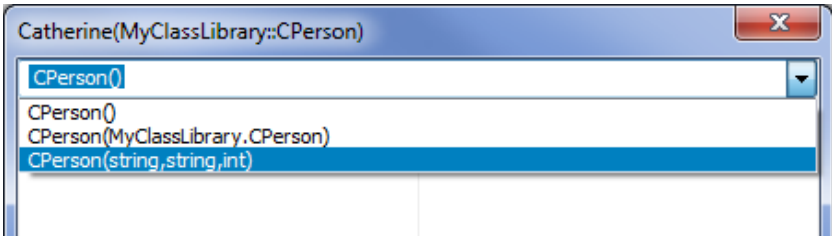
- [Workbench Setup](#) ^[1867]
- [Create & Delete Workbench Instances](#) ^[2177]
- [Invoke Methods](#) ^[2179]

15.4.2 Create & Delete Workbench Instances


This topic explains how to create and delete a workbench instance for any Class in your model.

Access **Project Browser Class context menu | Execution Analyzer | Create Workbench Instance of Class (Ctrl+Shift+J)**
Diagram Class context menu | Execution Analyzer | Create Workbench Instance of Class

Topics

Topic	Detail	See also
Naming the Workbench	<p>Select the Create Workbench Instance of Class option</p> <p>The Workbench dialog displays</p>  <p>Type in a name for the new instance (for example, Catherine); each instance name must be unique for the workbench.</p>	
Choosing a Constructor	<p>Click on the Create button</p> <p>The following dialog displays:</p>  <p>Select the constructor from the drop-down list.</p> <p>If you select a constructor with parameters, a set of fields displays in which you enter the value for each parameter (see <i>Enter Parameters</i>, below).</p> <p>If you do not define a constructor, or define a single constructor taking no</p>	

Topic	Detail	See also						
	<p>parameters, go straight to the row entitled <i>Invoke Workbench Instance</i>.</p>							
<p>Enter Parameters</p>	<p>In the dialog above, type any parameters required by the constructor:</p> <ul style="list-style-type: none"> • Literals as arguments <ul style="list-style-type: none"> • Text: abc or "abc" or "a b c" • Numbers: 1 or 1.5 • Objects as arguments <p>If an argument is not a literal then you can supply it in the list only if you have already created an instance of that type in the workbench; you do this by typing the name of the instance as the argument.</p> <p>The debugger checks any name entered in an argument against its list of workbench instances, and substitutes that instance in the actual call to the method.</p> • Strings as arguments <p>Surrounding strings with quotes is unnecessary as anything you type for a string argument becomes the value of the string; for example, the only time you should surround strings with quotes is in supplying elements of a string array, or where the string is equal to the name of an existing workbench instance.</p> <pre>"A b c" "a b \$ % 6 4" A b c d As 5 7) 2 === 4</pre> • Arrays as arguments <p>Enter the elements that compose the array, separated by commas:</p> <table border="0"> <tr> <td>Type</td> <td>Arguments</td> </tr> <tr> <td>String()</td> <td>one,two,three,"a book","a bigger book"</td> </tr> <tr> <td>CPerson()</td> <td>Tom,Dick,Harry</td> </tr> </table> 	Type	Arguments	String()	one,two,three,"a book","a bigger book"	CPerson()	Tom,Dick,Harry	
Type	Arguments							
String()	one,two,three,"a book","a bigger book"							
CPerson()	Tom,Dick,Harry							

Topic	Detail	See also
	<p>If you enter text that matches the name of an existing instance, surround it in quotes to avoid the debugger passing the instance rather than a string.</p> <pre>void SetName(string)</pre> 	
Invoke Workbench Instance	<p>Click on the Invoke button to commit the workbench instance.</p> <p>The instance then displays in the Workbench window, showing the variables of the instance in a hierarchy of the type and value of the instance and of any members.</p>	
Delete Workbench Instances	<p>You can delete instances using the Delete shortcut menu on any instance on the Workbench.</p> <p>If all instances are deleted, the debugger is shut down and the Workbench window is closed.</p>	

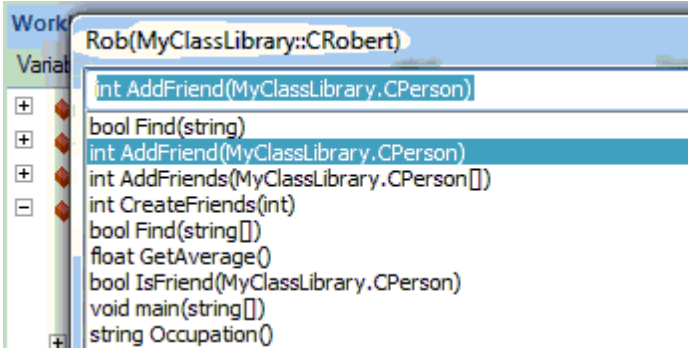
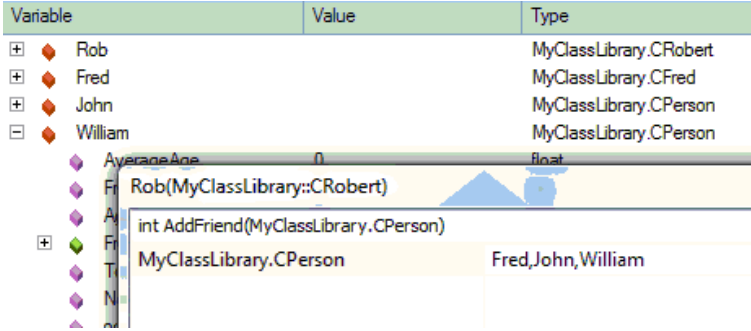
Learning Center topics

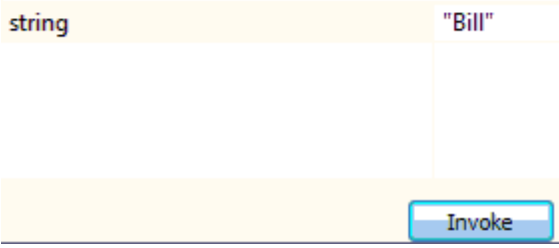

- (Alt+F1) | [Execution Analysis | Object Workbench | Creating Variables](#)

15.4.3 Invoke Methods

Topics

Topic	Detail	See also
Access	<p>On the Workbench window, right-click on the instance on which to execute a method, and select the Invoke context menu option.</p>	
Choose Method	<p>A list of methods for the type are presented in a dialog; select a method from the list and click on the Invoke button.</p> <p>Note that all methods listed are public; private methods are not available.</p>	

Topic	Detail	See also
		
<p>Supply Arguments</p>	<p>In the example, the instance or variable <i>Rob</i> has been created, of type <i>MyClassLibrary.CRobert</i>, and the method <i>AddFriends</i> invoked that takes an array of <i>CPerson</i> objects as its only argument.</p> <p>You now supply to it three other Workbench instances: <i>Fred</i>, <i>John</i> and <i>William</i>.</p> 	
<p>Arguments</p>	<p>In the dialog above, type any parameters required by the constructor:</p> <ul style="list-style-type: none"> • Literals as arguments <ul style="list-style-type: none"> • Text: abc or "abc" or "a b c" • Numbers: 1 or 1.5 • Objects as arguments <p>If an argument is not a literal then you can supply it in the list only if you have already created an instance of that type in the workbench; you do this by typing the name of the instance as the argument.</p> <p>The debugger checks any name entered in an argument against its list of workbench instances, and substitutes that</p>	

Topic	Detail	See also						
	<p>instance in the actual call to the method.</p> <ul style="list-style-type: none"> Strings as arguments <p>Surrounding strings with quotes is unnecessary as anything you type for a string argument becomes the value of the string; for example, the only time you should surround strings with quotes is in supplying elements of a string array, or where the string is equal to the name of an existing workbench instance.</p> <pre>"A b c" "a b \$ % 6 4" A b c d As 5 7) 2 === 4</pre> Arrays as arguments <p>Enter the elements that compose the array, separated by commas:</p> <table border="0"> <thead> <tr> <th>Type</th> <th>Arguments</th> </tr> </thead> <tbody> <tr> <td>String()</td> <td>one,two,three,"a book","a bigger book"</td> </tr> <tr> <td>CPerson()</td> <td>Tom,Dick,Harry</td> </tr> </tbody> </table> <p>If you enter text that matches the name of an existing instance, surround it in quotes to avoid the debugger passing the instance rather than a string.</p> <pre>void SetName(string) string "Bill"</pre>  	Type	Arguments	String()	one,two,three,"a book","a bigger book"	CPerson()	Tom,Dick,Harry	
Type	Arguments							
String()	one,two,three,"a book","a bigger book"							
CPerson()	Tom,Dick,Harry							
Invoke	<p>Having chosen the constructor and supplied any arguments, click on the Invoke button to create the variable.</p> <p>Output confirming this action is displayed in the Debug window.</p>	Run the Debugger 						

Learning Center topics

- (Alt+F1) | [Execution Analysis](#) | [Object Workbench](#) | [Invoke Methods](#)

15.5 Unit Testing

Enterprise Architect supports integration with unit testing tools in order to make it easier to develop good quality software.

In sequence:

- You download and install the NUnit and JUnit applications (JUnit - <http://www.junit.org/> NUnit - <http://www.nunit.org/index.php?p=home>) - Enterprise Architect does not include these applications in the installer
- Enterprise Architect helps you to create test Class stubs with the **JUnit** and **NUnit** transformations
- You define your test code within the Class stubs
- You set up and run a **test script** against any package
- All test results are automatically recorded inside Enterprise Architect

Learn more

- [JUnit Transformation](#)^[1702]
- [NUnit Transformation](#)^[1704]
- [Set Up Unit Testing](#)^[2183]
- [Add Testing Command](#)^[2183]
- [Run Unit Tests](#)^[2183]
- [Record Test Result](#)^[2183]

Learning Center topics

- (Alt+F1) | **Execution Analysis** | **xUnit Tests**

15.5.1 Set Up Unit Testing

This topic explains the actions you should take in setting up Unit Testing, after having downloaded and installed the JUnit and/or NUnit applications.

Topics

Topic	Detail	See also
Create Unit Test Stubs	<p>By using the JUnit or NUnit transformations and code generation you can create test method stubs for all of the public methods in each of your Classes.</p> <pre>(Test Fixture) public class CalculatorTest { (Test) public void testAdd() { } (Test)</pre>	<p>Unit Testing^[2183]</p> <p>JUnit Transformation^[1702]</p> <p>NUnit Transformation^[1704]</p> <p>Generate Source Code^[1768]</p>

Topic	Detail	See also
	<pre> public void testDivide() { } (Test) public void testMultiply() { } (Test) public void testSubtract() { } } </pre>	
<p>Define Test Cases</p>	<p>Write your unit test in the generated code stubs (either in Enterprise Architect or your preferred IDE).</p> <p>The following is an <i>NUnit</i> example in C#, although it could also be any other .NET language, or Java and JUnit.</p> <pre> (Test Fixture) public class CalculatorTest { (Test) public void testAdd() { Assert.AreEqual (1+1, 2); } (Test) public void testDivide() { Assert.AreEqual (2/2, 1); } (Test) public void testMultiply() { Assert.AreEqual (1*1, 1); } (Test) public void testSubtract() { Assert.AreEqual (1-1, 1); } } </pre> <p>Alternatively, if you have not performed an xUnit transformation, you can reverse engineer the above code into Enterprise Architect so that Enterprise Architect can record all test results against this Class.</p>	
<p>Compile Your Code</p>	<p>Ensure the source code being tested is compiled without errors, so that the test scripts can be run against it.</p>	
<p>Set up the Test Scripts</p>	<p>Set up the Test scripts against the required package, and then run the tests.</p>	<p>Add Testing Command¹⁸³⁴</p> <p>Run Unit Tests²¹⁸⁵</p>

Learning Center topics

- (Alt+F1) | [Execution Analysis](#) | [xUnit Tests](#) | [Set up Unit Tests](#)

15.5.2 Run Unit Tests

On running a test script you generate test results that are stored as Test Cases against the Classes being tested.

Access [Analyzer](#) | [Test](#) | [Run Test Script](#)

Topics

Topic	Detail	See also
Run Tests	<p>Select the appropriate package in the Project Browser.</p> <p>Select the Run Test Script option to run the test script you previously set up for that package, in the Execution Analyzer.</p>	Set Up Unit Testing ^[2183] Add Testing Command ^[1834]
View Results	<p>The results of xUnit tests are displayed in the Output window, identifying which tests have run and which of these have failed.</p> <p>The results also show which method failed and the file and line number the failure occurred at.</p> <p>Double-click on an error message; Enterprise Architect opens the editor to that line of code, enabling you to quickly find and fix the error.</p> <p>Enterprise Architect also records the run status of each test against the Class being tested; these are stored in the element Test Cases.</p> <p>A diagram containing the Class can be set to display these Test Cases, by exposing the test scripts compartment on the diagram elements.</p>	Record Test Results ^[2185] Show Test Scripts Compartment ^[2229]

Learning Center topics

- (Alt+F1) | [Execution Analysis](#) | [Run Unit Tests](#)

15.5.3 Record Test Results

Enterprise Architect is able to automatically record all results from tests through a testing script in Enterprise Architect.

Topics

Topic	Detail	See also
Process	<p>In order to use this feature, you must reverse engineer the test Class into the package containing your test script.</p> <p>Once your model contains your test Class, on the next run of the test</p>	Set Up Unit Testing ^[2183] Importing

Topic	Detail	See also
	<p>script Enterprise Architect adds Test Cases to the Class for each test method found; on this and all subsequent test runs all Test Cases are updated with the current run time and whether they passed or failed, as shown below:</p> <div data-bbox="472 483 769 855" style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <pre> Calculator Test + testAdd() : void + testDivide() : void + testMultiply() : void + testSubtract() : void test scripts Unit : (Pass) testAdd Unit : (Pass) testDivide Unit : (Pass) testMultiply Unit : (Fail) testSubtract </pre> </div> <p>The error description for each failed test is added to any existing results for that Test Case, along with the current date and time.</p> <p>Over time this provides a log of all test runs where each Test Case has failed, which can then be included in generated documentation, resembling the following:</p> <pre> Failed at 05- Jul - 2006 1: 02: 08 PM expected: <0> but was: <1> Failed at 28- Jun- 2006 8: 45: 36 AM expected: <0> but was: <2> </pre>	<p>Source Code^[1790]</p> <p>Run Unit Tests^[2185]</p>

15.6 Testpoint Management

Testpoint Management provides the facility to test and pass or fail application tasks, viewing test results in real time as the program executes and results are saved.

Tests	Id	Invariant	Pre	Post	Author	Status	Evals	Passes	Fails
CTrain									
Invariant					smeagher	✓	3302	3301	1
Passengers > 0						✓			
void Create()									
CTrain(CNetwork*, TTr)									
~CTrain()									
DWORD Disembark(int)		✓			smeagher	✓	6	6	
Pre-Condition									
PeopleOFF > 0						✓			
DWORD Embark(int)		✓			smeagher	✓	6	6	
Pre-Condition									
PeopleON > 0						✓			
DWORD Execute(CTrair)									
void Exit()									

The image shows a Testpoint run in progress. The example was produced using the Example Model that ships with Enterprise Architect

Topics

Topic	Detail	See also
Overview	<p>You define tests as Class invariants, and as pre/post conditions on operations.</p> <p>Testpoint design and management is performed on a Test Domain diagram, on which you define constraints and group them into sets, which in turn can form suites.</p> <p>The Testpoint interface is contextual, and when an object (Class, Test Cut, Test Set, Test Suite) is selected, the constraints for that object are displayed.</p> <p>In the Testpoint system, the rules that define task behaviors are defined within the model and do not form part of any code base, which has a number of advantages - changing a test condition in other systems can require that the application be re-built; in Enterprise Architect you could stop the run, change a constraint and start the run again without re-building the project.</p>	
Test Domain Diagram	<p>A Test Domain diagram is a particular diagram designed for use with the Testpoint facility.</p> <p>A Test Domain diagram provides specific elements that can aid in the logical composition of the Tests into Test Sets and Suites.</p>	<p>Test Domain Group^[72]</p> <p>Test Cut^[2198]</p>

Topic	Detail	See also
	<p>You can perform a Testpoint run on an entire Suite, a single Set or an individual Class.</p>	<p>Test Set ^[2198] Test Suite ^[2199] Combine Testpoints ^[2196]</p>
<p>Testpoint Class</p>	<p>Test conditions are always defined on a Class; invariants are created for the Class itself, and pre- and post-conditions are defined on its operations.</p> <p>These conditions can be aggregated later into sets, but the Classes are the building blocks upon which any sets you create are built.</p>	
<p>Constraints</p>	<p>The Testpoint facility is based on a programming-by-contract model. The Execution Analyzer uses constraints that you define for a Class and its methods:</p> <ul style="list-style-type: none"> • A Class invariant is evaluated by the Analyzer whenever a method called on an object of the Class completes • Pre-conditions defined for a method are evaluated when the method is first called; similarly, Post-conditions are evaluated (at the same time as any Class invariant) when the method completes <p>In the Testpoint facility, constraints are composed using member and local variables in expressions separated by operands; precedence can be achieved through the use of parentheses, which also permits the construction of complex expressions - you can specify any variable that would be in scope at the time.</p> <p>Elements and members of elements can also form part of the constraint; almost all equivalence operands are supported for primitive types and strings.</p>	<p>Testpoint Editor ^[2195]</p>
<p>Testpoint Constraints</p>	<p>An <i>invariant</i> defines the rules or constraints of a Class; it is expected that no objects of the Class nor its methods can break this constraint, thus preserving the state of the object.</p> <p>In the Testpoint facility, constraints are composed in much the same way as conditions are written in code; they rely on and are tied ultimately to the named attributes of the Class.</p> <p>The Execution Analyzer can take a single invariant defined for a Class and test it against every invocation of any method on any instance of that Class; the results can be seen during real time in the Testpoints window</p>	<p>The Testpoints Window ^[2189] Class Invariant Dialog ^[2193]</p>
<p>Trace Statements</p>	<p>Trace statements are used to output messages during the execution of a program; the Execution Analyzer provides this facility in:</p> <ul style="list-style-type: none"> • Debugging through the use of Tracepoints and 	

Topic	Detail	See also
	<ul style="list-style-type: none"> Testing through the use of Testpoints <p>In Testpoints, a trace statement can be associated with a pre- or post- condition; whenever the condition is evaluated, the trace statement is output.</p> <p>You can also group trace statements by level; when you perform a Testpoint run, you can choose the level of trace to be output.</p> <p>Output can be directed to the Testpoints tab of the Output window or to a file, as configured in the Analyzer script for the parent package.</p> <p>In debugging, trace statements are associated with code breakpoints.</p> <p>You can include the values of variables in trace statements, by prefixing the variable name with a special token that lets the debugger know it is not a part of the text; the available tokens are:</p> <ul style="list-style-type: none"> \$ - when the variable is to be interpreted or printed as a string @ - when the variable is a primitive type (int, double, char) <p>Variables can be qualified - that is, the members of a variable can be printed.</p> <p>You separate member names with a period character ('.')</p> <p><i>The value of the station name = \$station.name</i> <i>Departing passenger count = @passenger.count</i></p>	<p>The Output Window^[164]</p> <p>Managing Scripts^[1828]</p>

Learning Center topics

- (Alt+F1) | **Execution Analysis | Testpoints | Introducing Testpoints**

15.6.1 The Testpoints Window

The Testpoints window lists the test and trace items for the testpoint object currently selected on a diagram or in the Project Browser.

You can execute, stop and filter the test runs using the window toolbar.

Access **Analyzer | Test | Testpoint Manager**

Use to

- Review tests and traces for testpoint objects

Reference

Column	Usage	See also
Tests	<p>Displays the name of the selected Testpoint object and the hierarchy of objects beneath it.</p> <p>The selected object can be a:</p> <ul style="list-style-type: none"> • Class • Operation • TestCut • TestSet or • TestSuite 	<p>Test Cut^[2198]</p> <p>Test Set^[2198]</p> <p>Test Suite^[2199]</p>
Id	<p>If the test object represents an operation, an icon is displayed here when the test run commences, indicating a successful binding of that method's constraints.</p>	
Invariant	<p>A pencil icon in this column indicates that a Class Invariant is defined for this Class.</p> <p>Double-click on the item to add a constraint (no pencil icon) or edit the existing constraints (pencil icon) on the Class Invariant dialog.</p>	<p>Class Invariant dialog^[2193]</p>
Pre	<p>A pencil icon in this column indicates that a pre-condition is defined for this method.</p> <p>Double-click on the item to add a constraint (no pencil icon) or edit the existing constraints (pencil icon) on the Operation Constraints dialog.</p>	<p>Operation Constraints dialog^[2193]</p>
Post	<p>A pencil icon in this column indicates that a post-condition is defined for this method.</p> <p>Double-click on the item to add a constraint (no pencil icon) or edit the existing constraints (pencil icon) on the Operation Constraints dialog.</p>	<p>Operation Constraints dialog^[2193]</p>
Author	<p>Displays the username of the author of the constraints.</p>	
Status	<p>During a test run, indicates the last evaluated status of the test - either pass or fail.</p>	
Evals	<p>During a test run, indicates the number of times the Execution Analyzer has evaluated this test.</p>	
Passes	<p>During a test run, indicates the number of times the test passed.</p>	

Column	Usage	See also
Fails	During a test run, indicates the number of times the test failed.	
Last Run By	Displays the username of the last person to run this test.	
Last Run Date	Displays the date this test was last evaluated.	
Last Run Result	If the test has previously been run, displays the result of the last run.	
Parent Collections	Lists any parent collections that include the selected object as part of their design. Double-click this collection to make it the selected object in the left pane.	

Learn more

- [The Testpoints Window Toolbar](#)^[2191]

Learning Center topics

- (Alt+F1) | **Execution Analysis | Testpoints |**
 - **Add Class**
 - **Add Test Cut**
 - **Add Test Set**
 - **Add Test Suite**

15.6.1.1 *The Testpoints Window Toolbar*

The Testpoint window toolbar enables you to execute, stop and filter the output of a Test run.







Access **Analyzer | Test | Testpoint Manager**

Use to

- Manage the execution of a test run

Reference



Action	Usage	See also
Text field	Display the currently-selected test point object.	
	Execute the test run.	
	Stop the test run.	
	Toggle between showing all items and showing only those items that have constraints.	
	Toggle between showing all items and showing only operations that have been selected; this button is only enabled when a Testcut is selected. When a Testcut is selected, each of the operations of its Class are displayed with a checkbox; you use this checkbox to select the operations of the Testcut.	
	Display the Test Option menu, providing the following options: <ul style="list-style-type: none"> • All - Run all tests: include all tests and all trace statements • Testpoints Only - Run tests only: exclude trace statements • Tracepoints Only - Run trace statements only: exclude tests • Trace Level menu: <ul style="list-style-type: none"> • Disabled - All trace statements are excluded • Level 1 - Level 1 trace statements are included • Level 2 - Level 1 and 2 trace statements are included • Level 3 - All trace statements are included • Trace Output menu: <ul style="list-style-type: none"> • Prefix With Function Call - Prefix all trace output lines with the executing function name 	
	Display The Testpoints Window help topic.	

Learn more

- [Testpoint Management](#)^[2187]

15.6.1.2 Class Invariant Dialog

You use the Class Invariant dialog to define or edit the invariant constraint on a Class that is subject to Testpoint tests. A Class invariant is evaluated by the Analyzer whenever a method called on an object of the Class completes.

Access Double-click on the Class item in the Testpoints window

Reference

Field	Usage	See also
Name	Defaults to Invariant . Leave this value or, if required, overtype it with your preferred name.	
Constraints	Click on the (...) button. The Testpoint Editor dialog displays, listing the conditions currently set on the operation; you can create, delete and re-sequence the conditions on this dialog.	Testpoint Editor ^[2195]
Action on Fail	Click on the drop-down arrow and select from the three options: <ul style="list-style-type: none"> • Continue - ignore the failure • Break execution - halt the run and display the Stack trace • Disable on fail - do not execute the test again 	
OK	Click on this button to apply your changes.	

Learn more

- [The Testpoints Window](#)^[2189]

Learning Center topics

- (Alt+F1) | [Execution Analysis | Testpoints | Edit Invariant](#)

15.6.1.3 Operation Constraints Dialog

You use the Operation Constraints dialog to define or edit the pre-conditions and/or post-conditions on an operation in a Testpoint Class that is subject to Testpoint tests.

Pre-conditions defined for a method are evaluated when the method is first called; post-conditions are evaluated (at the same time as any Class Invariant) when the method completes.

Access Double-click on the operation item in the Testpoints window

Reference

The dialog contains two sets of the following fields; one set for pre-conditions and one set for post-conditions:

Field	Usage	See also
Constraint	Click on the (...) button. The Testpoint Editor dialog displays, listing the conditions currently set on the operation; you can create, delete and re-sequence the conditions on this dialog.	Testpoint Editor ^[2195]
Action on Fail	Click on the drop-down arrow and select from the three options: <ul style="list-style-type: none"> • Continue - ignore the failure • Break execution - halt the run and display the Stack trace • Disable on fail - do not execute the test again 	

Testpoint Trace:

The dialog also contains fields for Testpoint Trace statements. Testpoints differ from constraints, being statements that can return the values of variables (similar to a *Debug.Print* or *Session.output* statement). They are not set in the code, but stored in the model.

Field	Usage	See also
Trace	(Disabled if the Level field is set to Disabled). View the current traces and/or, if necessary, type in a new trace statement. If there are several trace statements, you can list the statements in a vertical list rather than a comma-separated string - click on the Browse button.	Testpoint Management ^[2188]
Level	Select the required level of trace statement to be output: <ul style="list-style-type: none"> • Disabled - no statement output • Level 1 - Only Level 1 statements output • Level 2 - Both Level 1 and Level 2 statements output • Level 3 - Levels 1, 2, and 3 statements output 	

Learn more

- [The Testpoints Window](#) ^[2189]

Learning Center topics

- (Alt+F1) | **Execution Analysis | Testpoints |**
 - **Edit Pre Condition**

- **Edit Post Condition**

15.6.1.4 Testpoint Editor

You identify the Testpoint constraints to use on the Class Invariant dialog and the Operation Constraints dialog, but both dialogs call the Testpoint Editor dialog to enable you to:

- List the existing conditions in a vertical list rather than a comma-separated string (**Browse** button)
- Delete a condition applied to the invariant (**Delete** icon)
- Create a new condition (**New** icon, then type the condition in)
- Change the order in which the conditions are applied (**Move Item Up**, **Move Item Down** icons)

Constraint Composition:

Constraints are composed using member and local variables in expressions separated by operands such as:

- /
- *
- %
- +
- -
- >
- >=
- <
- <=
- OR
- AND
- (
-)

The following is an example Conditional expression that returns a Boolean True/False:

```
( ( x%2 ) > 1 ) AND ( y >= 10 )
```

Variables can be scoped by either '.' or '->'

```
c. Name : c. Names[ 0 ] of an array as c. Names[ 0 ]. Sur name = " S m i t h "
```

There are two types of operators you can use on constraints:

- Logical operators AND and OR can be used to combine constraints
- Equivalence operators '=' and '!=' can be used to define the conditions of a constraint

All constraints are subject to an AND operation unless otherwise specified; you can use the OR operation on them instead, for example:

```
Locat ion=0 OR
Locat ion=1 AND
Depar ting. Name! =Cent ral
```



```
( a AND b)
(a. b AND ( b. a OR c. a ) )
```

Below are some examples of using the equivalence operators:

```
m_val ue! =" t r u e"
m_val ue <> 0

m_val ue[ 10] =" j o h n"
m_val ue[ 0] [ 1] =2
```

Notes

- Quotes around strings are optional; the comparison for strings is always case-sensitive in determining the truth of a constraint

Learn more

- [Testpoint Management](#)^[2188]

Learning Center topics

- (Alt+F1) | [Execution Analysis | Testpoints | About Constraints](#)
- (Alt+F1) | [Execution Analysis | Testpoints | About Statements](#)

15.6.1.5 Combine Testpoints

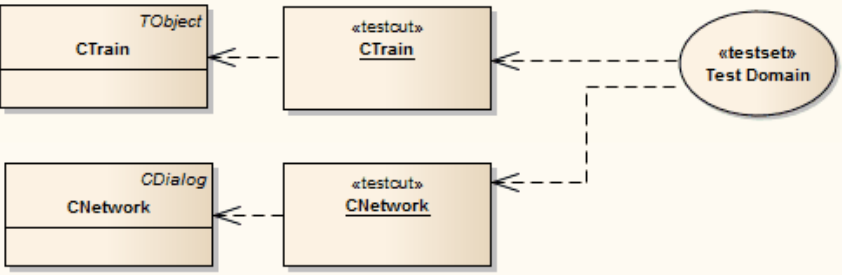
Testpoints are applied to a Class. Where functionality requires tests across multiple Classes, you can group the tests together to allow a single test run across the group of Classes.

You group the tests on a Test Domain diagram, either:

- Manually using the Test Domain Toolbox pages and Project Browser, or
- Automatically from the results of using the debugger and Record & Analyze window

Topics

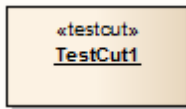
Topic	Detail	See also
Create Test Domain diagram manually	<p>Create a new Test Domain diagram under the required package, and drag a Test Cut element from the Test Domain Toolbox page onto the diagram; give the element an appropriate name, such as the name of the Class it acts on.</p> <p>Drag the Class containing the tests from the Project Browser onto the diagram, and select the 'paste as simple link' option on the Paste Element dialog; create a Dependency connector between the Test Cut element and the Class.</p> <p>(When you select the Test Cut element on the diagram when the Testpoints window is open, the operations of the Class are listed in the window, each followed by a checkbox; you then select the checkbox against each Class operation to include in the Test Cut).</p> <p>Drag and connect further Test Cut and Class element pairs on the diagram as appropriate.</p> <p>Drag a Test Set element from the Toolbox onto the diagram, and create a</p>	<p>Test Domain Group^[721]</p> <p>Test Cut^[2198]</p> <p>Test Set^[2198]</p>

Topic	Detail	See also
	<p>Dependency relationship between this element and each of the Test Cut elements; the resulting diagram should resemble the following:</p>  <pre> classDiagram class TestDomain["«testset» Test Domain"] class CTrain["«testcut» CTrain"] class CNetwork["«testcut» CNetwork"] class CTrainObj["TObject CTrain"] class CNetworkObj["CDialog CNetwork"] TestDomain -.-> CTrain TestDomain -.-> CNetwork CTrain -.-> CTrainObj CNetwork -.-> CNetworkObj </pre> <p>If you want to further combine tests, set up additional Test Set hierarchies and then drag a Test Suite element onto the diagram from the Toolbox; create a Dependency relationship between this element and each of the Test Set elements.</p> <p>When you open the Testpoints window, you can select any element on the diagram to display that element and its hierarchy in the window; you can then execute the test run on your selection.</p> <p>If the selection checkboxes do not display, select the Testcut element on the diagram again.</p>	<p>Test Suite ^[2199]</p> <p>The Testpoints Window ^[2189]</p>
<p>Generate Test Domain diagram from Sequence Record</p>	<p>To generate a Testpoint Domain diagram automatically:</p> <ul style="list-style-type: none"> • Build and debug your code successfully • Work through the process of recording the stack trace to generate a Sequence diagram • When you have completed recording (whether or not you display the Sequence diagram), click on the Testpoints icon in the Record & Analyze window toolbar • Open the Testpoints window, select any element on the diagram to display that element and its hierarchy in the window, and execute the test run on your selection <p>You cannot automatically generate a Test Suite element from the Sequence record, only Test Sets, Test Cuts and Classes.</p>	<p>Recording Sequence Diagrams ^[2147]</p> <p>Recorder Toolbar ^[2163]</p> <p>The Testpoints Window ^[2189]</p>

Learning Center topics

- (Alt+F1) | **Execution Analysis | Testpoints | Add Test Cut**
- (Alt+F1) | **Execution Analysis | Testpoints | Add Test Set**
- (Alt+F1) | **Execution Analysis | Testpoints | Add Test Suite**

15.6.1.5.1 Test Cut



Description

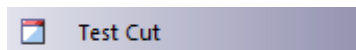
A Test Cut element is a stereotyped Object element, used internal to Enterprise Architect for defining test sets using the Testpoint code testing facilities.

A task, such as 'Print', might involve operations on different Classes. In order to create a 'Print' test, you would want to include only the 'Print' operations of these Classes and exclude any other operations.

A Test Cut enables you to capture only the operations that represent the behavior (in this case, 'Print') defined for a single Class. You might then place the Test Cut from each of several Classes into a single task as a Test Set.

When you drag a Test Cut element onto a Test Domain diagram, you create a Dependency relationship with the required Class element. As a result, when you select the *Test Cut* element on the Testpoints Window, the operations of the *Class* are listed in the window, each with a checkbox. You then select the checkbox against each Class operation to include in the Test Cut.

Toolbox icon



Learn more

- [Test Set](#)^[2198]
- [The Testpoints Window](#)^[2189]
- [Combine Testpoints](#)^[2196]

Learning Center topics

- (Alt+F1) | **Execution Analysis | Testpoints | Add Test Cut**

15.6.1.5.2 Test Set

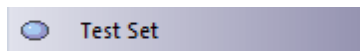


Description

A *Test Set* element is a stereotyped Use Case element used to aggregate one or more groups of methods (Test Cuts), which perhaps span multiple Classes, into a single task. Test Sets can also be aggregated into Test Suites.

You link the Test Cut elements to the Test Set using Dependency connectors.

Toolbox icon



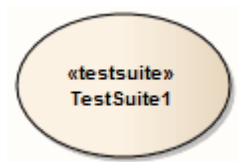
Learn more

- [Test Cut](#)^[2198]
- [Test Suite](#)^[2199]
- [Combine Testpoints](#)^[2196]

Learning Center topics

- (Alt+F1) | [Execution Analysis](#) | [Testpoints](#) | [Add Test Set](#)

15.6.1.5.3 Test Suite

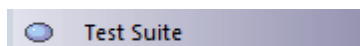


Description

A *Test Suite* element is a stereotyped Use Case element, used to aggregate one or more groups of tasks (Test Sets).

You link the Test Set elements to the Test Suite using Dependency connectors.

Toolbox icon



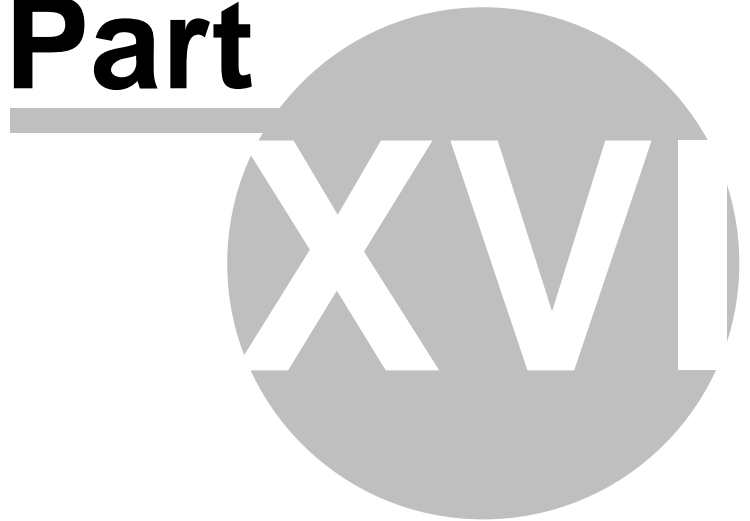
Learn more

- [Test Set](#)^[2198]
- [Combine Testpoints](#)^[2196]

Learning Center topics

- (Alt+F1) | [Execution Analysis](#) | [Testpoints](#) | [Add Test Suite](#)

Part



16 Testing



This section discusses quality control in your project with Enterprise Architect, describing:

Topic	Link
<p>Model Validation Checks - Used to check UML elements, diagrams or packages against known UML rules (identified in configuring validation) and constraints defined within the model, using the Object Constraint Language (OCL).</p>	<p>Model Validation Checks ^[2203]</p>
<p>Testing - Enterprise Architect enables you to create and manage test scripts for model elements, covering unit, integration, scenario, system and acceptance tests.</p>	<p>Testing ^[2211]</p>
<p>Testpoint Management - provides the facility to pass or fail application tasks, viewing test results in real time as the program executes and results are saved.</p>	<p>TestPoint Management ^[2187]</p>
<p>Integration with the unit testing tools JUnit and NUnit.</p>	<p>Unit Testing ^[2183]</p>

16.1 Model Validation

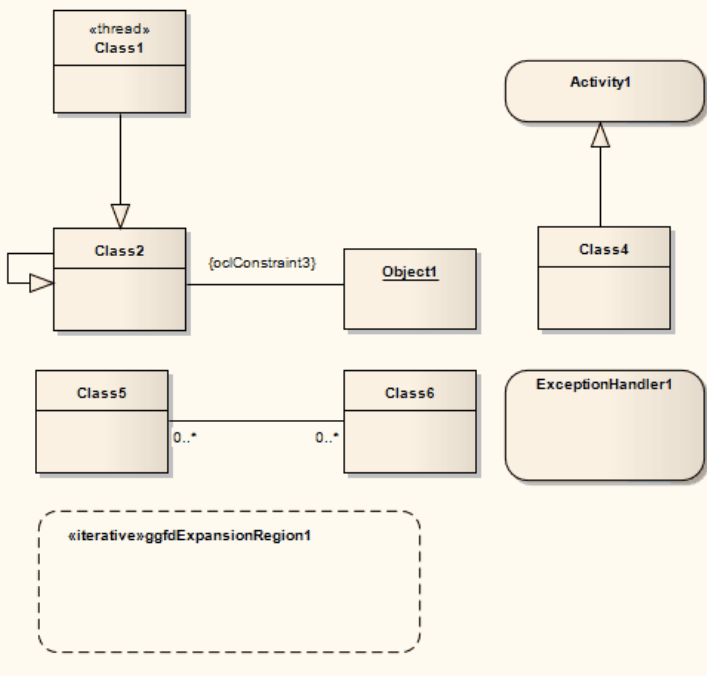


You use Model Validation to check UML models against known **UML rules** (which you identify in **configuring validation**) as well as any constraints defined within the model, using the Object Constraint Language (OCL). You can run Model Validation against a single UML element, a diagram or an entire package.

Access **Project | Model Validation | Validate Selected (Ctrl + Alt + V)**

Topics

Topic	Detail	See also
Validating	Validating a UML: <ul style="list-style-type: none"> • Element validates the element and its children, features (attributes and operations) and relationships (connectors) • Diagram validates the diagram itself (for correctness) as well as any elements and connectors within the diagram • Package validates the package and all sub-packages, elements, connectors and diagrams within it 	
Example - Model Violation	The following UML diagram contains several basic violations:	

Topic	Detail	See also
	 <p>If you run Model Validation on this diagram, Enterprise Architect displays the following violations in the Output window:</p> <ul style="list-style-type: none"> • It contains a UML ExpansionRegion (<i>ExpansionRegion1</i>) that is missing its child input ExpansionNode • It contains an invalid self-generalization on <i>Class2</i> (UML elements cannot be self-generalized) • It contains an OCL violation for the anonymous Association (between <i>Class2</i> and <i>Object1</i>) • It contains a UML ExceptionHandler (<i>ExceptionHandler1</i>) that is missing its child input ObjectNode 	

How to

To use Model Validation

Step	Action	See also
1	Select the package, diagram or element either from the Project Browser or within an open diagram.	
2	Select the Validate Selected menu option. Enterprise Architect performs the validation, and displays the results in the Output	Output Window ^[164]

Step	Action	See also
	<p>window (if the Output window does not automatically display, select the View System Output menu option).</p> <p>While performing the validation, Enterprise Architect also displays a progress window containing the Cancel Validation button, which enables you to cancel the validation process at any time.</p> <p>Alternatively, select the Project Model Validation Cancel Validation menu option.</p>	

Notes

- If you double-click on an error in the Output window, you select the diagram element that the error message refers to

Learn more

- [UML rules](#) ^[2206]
- [Configuring Model Validation](#) ^[2205]

16.1.1 Configure Model Validation

Use the Model Validation Configuration dialog to enable and disable the rules that are run with the model validator. You can define additional rules in this dialog from any additional Add-Ins that might be installed besides Enterprise Architect.

Access **Project | Model Validation | Configure**

Topics

Topic	Detail	See also
Usage	<p>Click on the checkbox against each Validation Rule to apply in performing a model validation.</p> <p>When you perform a validation, each violation listed on the Output window has a violation ID of the format <i>MVRxxnnnn</i>, where:</p> <ul style="list-style-type: none"> • <i>MVR</i> stands for Model Validation Rule • <i>xx</i> is a hexadecimal number corresponding to the position of the validation rule in the Model Validation Configuration dialog, thus indicating which rule is applied and violated • <i>nnnn</i> is the number of the violation message <p>Therefore:</p>	<p>Model Validation ^[2203]</p>

Topic	Detail	See also
	<ul style="list-style-type: none"> • Messages with the ID <i>MVR01nnnn</i> indicate that the Element: Well-Formedness checkbox is selected and a violation of that rule has been detected • Messages with the ID <i>MVR0Annnn</i> indicate that the Feature: OCL Conformance checkbox (10th in order on the dialog, or <i>Ath</i> in hexadecimal) is selected and a violation of that rule has been detected 	

Notes

- To disable UML syntax ("*The requested connection is not UML compliant*"), select the **Tools | Options | Diagram** option, and in the General panel deselect the **Strict UML Syntax** checkbox

Learn more

- [Rules Reference](#) [2206]

Learning Center topics

- (Alt+F1) | **Testing | Model Validation | How to Configure**

16.1.2 Rules Reference

Topics

Topic	Detail	See also
Validation Rules	<p>Model Validation works against a set of validation rules, arranged in the following groups:</p> <ul style="list-style-type: none"> • (Element, Relationship, Feature, Diagram): Well-Formedness Checks whether or not an element, relationship, feature or diagram is well-formed; this group of rules includes checks such as whether the item is a valid UML item and whether a diagram contains valid elements within it • Element: Composition Checks whether or not a UML element contains valid children, whether it contains the right number of valid children, and whether or not the element is missing any required children 	<p>Well-Formedness <small>[2207]</small></p> <p>Element: Composition <small>[2208]</small></p> <p>Property Validity <small>[2208]</small></p>

Topic	Detail	See also
	<ul style="list-style-type: none"> (Element, Relationship, Feature): Property Validity Checks whether or not the item has the correct UML properties defined, and whether the properties contain incorrect or conflicting values (Element, Relationship, Feature): Custom Properties Validates an item against any defined constraints in OCL 	OCL Conformance [2209]

16.1.2.1 Well-Formedness

This group of rules checks whether or not an element, relationship, feature or diagram is well-formed. The rules includes checks such as whether the item is a valid UML item and whether a diagram contains valid elements within it.

Reported violations:

Violation ID	Description	Information
MVR010001	«Element» is not a valid UML Element	The element is not a recognized UML 2.3 element.
MVR050001	«Relationship» is not a valid UML Relationship	The relationship is not a recognized UML 2.3 relationship.
MVR050002	«Relationship» is not legal for «Start Element» --> «End Element»	The relationship between the given start and end elements is not valid for those elements.
MVR050003	«Parent Element»:isLeaf=true and cannot be generalized by «Child Element»	The Generalization relationship cannot exist between parent and child elements because the parent element is defined as a leaf element.
MVR050004	«Child Element»:isRoot=true and cannot generalize «Parent Element»	The Generalization relationship cannot exist between parent and child elements because the child element is defined as a root element.
MVR050005	«Element» cannot generalize self	The element cannot be self-generalized.
MVR0B0001	Statechart violation: «extended information»	The State diagram contains a UML violation; see the extended information for more information about the detected violation.

16.1.2.2 Element Composition

This group of rules checks whether or not a UML element contains valid children, whether it contains the right number of valid children, and whether or not the element is missing any required children.

Reported violations:

Error ID	Description	Information
MVR020001	«Element» is missing required child element «Child Element»	The element is missing a child element of type <i>Child Element</i> .
MVR020002	Invalid UML package child	The element cannot be a direct package child and must be a child of another element (for example: Ports must be children of other elements, and not direct UML package members).
MVR020003	Invalid child «Child Element name» («Child Element Type»)	The child element is invalid on the tested parent element.

16.1.2.3 Property Validity

This group checks whether or not an element, relationship or feature has the correct UML properties defined for it and whether they contain incorrect or conflicting values.

Reported violations:

Error ID	Description	Information
MVR030001	«Element»:«Property» property is undefined	The element property contains no value.
MVR030002	«Element»:«Property» property has invalid value: "«Value»"	The element property contains an invalid value.
MVR030003	«Element»:isLeaf=true and cannot be abstract	The element's <i>isLeaf</i> and <i>isAbstract</i> properties are both set to true , which is invalid.
MVR060001	«Relationship»:«Property» property is undefined	The relationship property contains no value.
MVR060002	«Relationship»:«Property» property has invalid value: "«Value»"	The relationship property contains an invalid value.

Error ID	Description	Information
MVR090001	Attribute/AssociationEnd mismatch, «Attribute»: «Mismatch description», ...	The given attribute has an <i>associationEnd</i> of the same name but they differ in the listed details.

Learn more

- [Properties Dialog](#)^[843] (Element)
- [Connector Properties](#)^[961]

16.1.2.4 OCL Conformance

This group validates an element, relationship or attribute against any defined constraints in the Object Constraint Language (OCL). OCL is used to describe expressions on UML models, and to express constraints free of side-effects. You can add OCL constraints to any element, relationship or attribute in Enterprise Architect.

Reported violations:

Error ID	Description	Information
MVR040001	OCL violation: «violated OCL»	The element violates the OCL constraint specified.
MVR070001	OCL violation: «violated OCL»	The relationship violates the OCL constraint specified.
MVR0A0001	OCL violation: «violated OCL»	The attribute violates the OCL constraint specified.

Topics

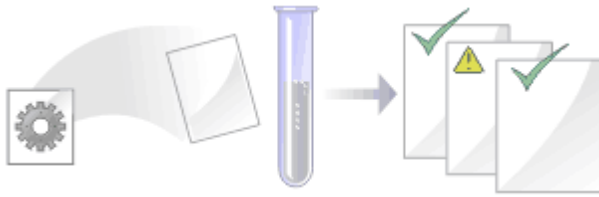
Topic	Detail	See also
Define OCL Constraints for an Element	<p>You can add OCL constraints to an element using the Properties dialog (Element Properties); select the Constraints page, click on the Type drop-down arrow and select OCL.</p> <p>To perform an OCL Validation, display the Model Validation Configuration dialog and select the Element: (OCL) Conformance checkbox; any OCL violations are recorded in the Model Validation tab of the Output window.</p>	<p>Model Validation Configuration^[2205]</p> <p>Model Validation^[2203]</p>
Define OCL Constraints for a Relationship	<p>You can add OCL constraints to a relationship using the Properties dialog (right-click and select the <type> Properties context menu option); select the Constraints page, click on the Type drop-down</p>	

Topic	Detail	See also
	<p>arrow and select OCL.</p> <p>To perform an OCL Validation, display the Model Validation Configuration dialog and select the Relationship: (OCL) Conformance checkbox; any OCL violations are recorded in the Model Validation tab of the Output window.</p>	
Define OCL Constraints for an Attribute	<p>You can add OCL constraints to a feature using the Properties dialog (Element Attributes); select the Constraints tab, click on the Type drop-down arrow and select OCL.</p> <p>To perform an OCL Validation, display the Model Validation Configuration dialog and select the Feature: (OCL) Conformance checkbox; any OCL violations are recorded in the Model Validation tab of the Output window.</p>	

Notes

- To have a valid OCL constraint, the syntax must be correctly formed; if the expression is not correct, Enterprise Architect displays a message stating that the OCL constraint is not valid

16.2 Testing



Topics

Topic	Detail	See also
Introduction to Testing	<p>Enterprise Architect enables you to create and manage test scripts for model elements, covering unit, integration, scenario, system and acceptance tests; this can include test cases generated from xUnit testing and Testpoint Management.</p> <p>You can import tests from other elements, generate them from scenarios, and generate test documentation and reports; you can also indicate the presence of tests on an element by displaying test information on the element in a diagram.</p> <p>Enterprise Architect enables you to attach arbitrarily complex tests to any model element; keeping the model elements and the testing documentation in one integrated model significantly improves the communication between the test-team and the software developers and architects.</p> <p>The detailed search facilities make it easy to find failing test cases, test cases not run and test cases that have been passed; using the testing and search capabilities, it is easy to navigate through the model and quickly locate problem spots, design flaws and other critical issues.</p> <p>Enterprise Architect is not only a UML Modeling environment, it is also a complete Test Management environment.</p>	xUnit Testing ^[2185] Testpoint Management ^[2187]
Basic Tasks	<p>Simple tasks that you might perform include:</p> <ul style="list-style-type: none"> • Open the Testing Workspace • Use the Test Details dialog 	The Testing Workspace ^[2213] The Test Details Dialog ^[2214]
Categories	<p>Typically, you create:</p> <ul style="list-style-type: none"> • Unit tests for things that are being built, such as Classes and components • Integration tests to test how components work together • System tests to ensure the system meets business requirements • Acceptance tests to test user satisfaction, and • Scenario tests to test the end-to-end suitability and 	Unit Tests ^[2215] Integration Testing ^[2216] System Testing ^[2218] Acceptance testing ^[2220] Scenario testing ^[2221]

Topic	Detail	See also
	functionality of the application	
Using Tests	<p>Other tasks that you might perform when working with tests include:</p> <ul style="list-style-type: none"> • Import a scenario as a test • Move or copy tests between test types • Import a test from other elements • Import a responsibility or constraint as a test • Create a maintenance item from a test • Generate a Test Details report • Show test script compartments • Create test documentation 	<p>Import a Scenario as a Test ^[2224]</p> <p>Move or Copy Tests Between Test Types ^[2223]</p> <p>Import a Test from Other Elements ^[2225]</p> <p>Import a Responsibility or Constraint as a Test ^[2227]</p> <p>Create a Maintenance Item from a Test ^[2228]</p> <p>Generate Test Details Report ^[2343]</p> <p>Show Test Script Compartments ^[2229]</p> <p>Create Test Documentation ^[2230]</p>

Notes

- Most of the tasks identified above relate to a tests for a single element
- You can make a set of tests available to a number of elements by performing the above tasks on a **Test Case** element and then associating that Test Case with each of the other elements; the Test Case element also helps to make tests more visible in diagrams, the Project Browser, windows and searches

Learn more

- [Test Case](#) ^[1677]
- [xUnit Testing](#) ^[2183]
- [Testpoint Management](#) ^[2187]

Learning Center topics

- (Alt+F1) | **Execution Analysis | Testing | Testing**

16.2.1 The Testing Workspace

The Testing window, or Workspace, provides a quick and convenient method of working with element tests. When you select an element in a diagram or in the Project Browser, if the Testing window is visible the lists of tests for that element are loaded ready for modification or addition.

Access [View | More Element Tools | Testing](#) (Alt + 3)

Topics

Topic	Detail	See also
Using the Testing Workspace	<p>The window has two formats - Item mode and List mode respectively.</p> <p>To toggle between the modes, click on the Show/Hide Properties button in the window toolbar.</p> <p>To add new items, click on the New icon in the window toolbar; in:</p> <ul style="list-style-type: none"> • Item mode, this clears the fields for new data • List mode, this displays the Test Details dialog, which you complete in the same way as the Testing window in Item mode <p>By clicking on the Auto button in Item mode or on the Test details dialog, you can apply an automatic naming/numbering nomenclature that you have previously defined. If you already have some text in the Name field, it is over-written by the auto-counter text.</p> <p>There are five tabs along the base of the window; one for each of the following types of testing:</p> <ul style="list-style-type: none"> • Unit testing • Integration testing • System testing • Acceptance testing • Scenario testing <p>The asterisk on a tab indicates that the tab contains saved information; if there is no information for a type of test, or the information has not yet been saved, its tab has no asterisk.</p> <p>The tabs toggle between Item mode and List mode independently.</p>	<p>Test Dialog^[2214]</p> <p>Use Auto Naming and Auto Counters^[799]</p> <p>Unit testing^[2215]</p> <p>Integration testing^[2216]</p> <p>System testing^[2218]</p> <p>Acceptance testing^[2220]</p> <p>Scenario testing^[2221]</p>

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of

Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

- The Testing window can be docked to the application workspace

Learn more

- [Permission List](#)^[264]

Learning Center topics

- (Alt+F1) | **Execution Analysis | Testing | Testing Workspace**

16.2.2 The Test Details Dialog

The Test Details dialog opens to edit tests when the Testing window is in List mode.

Double-click on an existing test case or click on the **New** icon in the window toolbar. The Test Details dialog displays.

Complete the dialog in the same way as you complete the Testing window in Item mode, for each category of test.

Access **View | More Element Tools | Testing (Alt + 3) > List mode (:Show/Hide Properties if necessary): New toolbar icon, or**
View | More Element Tools | Testing (Alt + 3) > List mode (:Show/Hide Properties if necessary): double-click item for editing

Notes

- Add multiple test cases in one batch using the **New** and **Apply** buttons
- You can format the text in the Description, Input, Acceptance Criteria and Results tabs using the Notes toolbar at the top of the field; this text is also reflected in the Notes window, but cannot be edited there
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

Learn more

- [Unit Testing](#)^[2215]
- [Integration Testing](#)^[2216]
- [System Testing](#)^[2218]
- [Acceptance Testing](#)^[2220]
- [Scenario Testing](#)^[2221]
- [Notes Toolbar](#)^[979]

16.2.3 Unit Testing

Use Unit Testing to test Classes, Components and other elements as programmers build them.

Access [View | More Element Tools | Testing > Unit](#)

The Unit testing tab displays in the Testing window by default.


If any **Unit tests** exist, the first Test item for the element is shown in the Testing window in Item mode, and all items are listed either in the panel to the left of the window, or in the window in List mode.

Reference

Field	Usage	See also
Test	Specify the name of the test.	
Auto	As an alternative to typing in the name, click on this button to insert predefined auto-counter text. If you already have some text in the Test field, it is over-written by the auto-counter text.	Use Auto Naming and Auto Counters ^[799]
Status	Click on the drop-down arrow and select the current status of the test.	
Type	Click on the drop-down arrow and select the type of test.	
Run By	Click on the drop-down arrow and select the name of the person who ran the test.	
Checked By	Click on the drop-down arrow and select the name of the person who checked the test run.	
Last Run	Click on the drop-down arrow and select the date on which the test was last run.	
Description	Type a description of the test; you can format the text using the Notes toolbar at the top of the field. This text is also reflected in the Notes window, but cannot be edited there.	Notes Toolbar ^[979]
Input	Type in the input data.	
Acceptance	Type the acceptance or test success conditions.	

Field	Usage	See also
Criteria		
Results	Type the results of the last test.	

Topics

Topic	Detail	See also
Editing Unit Tests	<p>To edit existing Unit Test items for this element, either:</p> <ul style="list-style-type: none"> • Click on the item in the left-hand panel in Item mode • Double-click on the item in List mode to display the Test Details dialog, or • Click on the required item in the <i>Testing</i> folder in the Element Browser window (if this window is not displayed, click on the  icon in the Testing window toolbar); Unit Test item icons have a U in the bottom right corner <p>To change the element for which to create test items, click on the required element in the Project Browser.</p>	<p>Test Details Dialog^[2214]</p> <p>Element Browser^[876]</p>

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

16.2.4 Integration Testing



Use Integration Testing to test how the constructed components work together.

Access [View | More Element Tools | Testing > Integration](#)


Open a diagram or the Project Browser and select the required element, then click on the Integration tab.



If any Integration tests exist, the first Test item for the element is shown in the Testing window in Item mode, and all items are listed either in the panel to the left of the window, or in the window in List mode.

Reference

Field	Usage	See also
Test	Specify the name of the test.	
Auto	As an alternative to typing in the name, click on this button to insert predefined auto-counter text. If you already have some text in the Test field, it is over-written by the auto-counter text.	Use Auto Naming and Auto Counters 
Status	Click on the drop-down arrow and select the current status of the test.	
Type	Click on the drop-down arrow and select the type of test.	
Run By	Click on the drop-down arrow and select the name of the person who ran the test.	
Checked By	Click on the drop-down arrow and select the name of the person who checked the test run.	
Last Run	Click on the drop-down arrow and select the date on which the test was last run.	
Description	Type a description of the test; you can format the text using the Notes toolbar at the top of the field. This text is also reflected in the Notes window, but cannot be edited there.	Notes Toolbar 
Input	Type in the input data.	
Acceptance Criteria	Type the acceptance or test success conditions.	
Results	Type the results of the last test.	

Topics

Topic	Detail	See also
Editing Integration Tests	To edit existing Integration Test items for this element, either: <ul style="list-style-type: none"> Click on the item in the left-hand panel in Item mode 	Test Details Dialog  Element Browser

	<ul style="list-style-type: none"> • Double-click on the item in List mode to display the Test Details dialog, or • Click on the required item in the Testing folder in the Element Browser window (if this window is not displayed, click on the  icon in the Testing window toolbar); Integration Test item icons have a I in the bottom right corner <p>To change the element for which to create test items, click on the required element in the Project Browser.</p>	
--	--	---

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

16.2.5 System Testing


Use System Testing to test that the system performs the right business functions correctly.

Access [View](#) | [More Element Tools](#) | [Testing > System](#)

Open a diagram or the Project Browser and select the required element. Click on the System tab.


If any System Tests exist, the first Test item for the element is shown in the Testing window in Item mode, and all items are listed either in the panel to the left of the window, or in the window in List mode.

Reference

Field	Usage	See also
Test	Specify the name of the test.	
Auto	As an alternative to typing in the name, click on this button to insert predefined auto-counter text. If you already have some text in the Test field, it is over-written by the auto-counter text.	Use Auto Naming and Auto Counters 
Status	Click on the drop-down arrow and select the current status of the test.	
Type	Click on the drop-down arrow and select the type of test.	

Field	Usage	See also
Run By	Click on the drop-down arrow and select the name of the person who ran the test.	
Checked By	Click on the drop-down arrow and select the name of the person who checked the test run.	
Last Run	Click on the drop-down arrow and select the date on which the test was last run.	
Description	Type a description of the test; you can format the text using the Notes toolbar at the top of the field. This text is also reflected in the Notes window, but cannot be edited there.	Notes Toolbar [979]
Input	Type in the input data.	
Acceptance Criteria	Type the acceptance or test success conditions.	
Results	Type the results of the last test.	

Topics

Topic	Detail	See also
Editing System Tests	<p>To edit existing System Test items for this element, either:</p> <ul style="list-style-type: none"> Click on the item in the left-hand panel in Item mode Double-click on the item in List mode to display the Test Details dialog, or Click on the required item in the <i>Testing</i> folder in the Element Browser window (if this window is not displayed, click on the  icon in the Testing window toolbar); System Test item icons have a Sy in the bottom right corner <p>To change the element for which to create test items, click on the required element in the Project Browser.</p>	Test Details Dialog [2214] Element Browser [876]

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

16.2.6 Acceptance Testing

Use Acceptance Testing to ensure that users are satisfied with the system.

Access [View](#) | [More Element Tools](#) | [Testing > Acceptance](#)

Open a diagram or the Project Browser and select the required element. Click on the Acceptance tab.


If any Acceptance Tests exist, the first Test item for the element is shown in the Testing window in Item mode, and all items are listed either in the panel to the left of the window, or in the window in List mode.

Reference

Field	Usage	See also
Test	Specify the name of the test.	
Auto	As an alternative to typing in the name, click on this button to insert predefined auto-counter text. If you already have some text in the Test field, it is over-written by the auto-counter text.	Use Auto Naming and Auto Counters ^[799]
Status	Click on the drop-down arrow and select the current status of the test.	
Type	Click on the drop-down arrow and select the type of test.	
Run By	Click on the drop-down arrow and select the name of the person who ran the test.	
Checked By	Click on the drop-down arrow and select the name of the person who checked the test run.	
Last Run	Click on the drop-down arrow and select the date on which the test was last run.	
Description	Type a description of the test; you can format the text using the Notes toolbar at the top of the field. This text is also reflected in the Notes window, but cannot be	Notes Toolbar ^[979]

Field	Usage	See also
	edited there.	
Input	Type in the input data.	
Acceptance Criteria	Type the acceptance or test success conditions.	
Results	Type the results of the last test.	

Topics

Topic	Detail	See also
Editing Acceptance Tests	<p>To edit existing Acceptance Test items for this element, either:</p> <ul style="list-style-type: none"> Click on the item in the left-hand panel in Item mode Double-click on the item in List mode to display the Test Details dialog, or Click on the required item in the Testing folder in the Element Browser window (if this window is not displayed, click on the  icon in the Testing window toolbar); Acceptance Test item icons have a A in the bottom right corner <p>To change the element for which to create test items, click on the required element in the Project Browser.</p>	Test Details Dialog ^[2214] Element Browser _[876]

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

16.2.7 Scenario Testing



Use Scenario Testing to test the application with real-world situations and scenarios. An end-to-end test of all functions.

Access [View | More Element Tools | Testing > Scenario](#)


Open a diagram or the Project Browser and select the required element. Click on the Scenario tab.

If any Scenario Tests exist, the first Test item for the element is shown in the Testing window in Item mode, and all items are listed either in the panel to the left of the window, or in the window in List mode.

Reference

Field	Usage	See also
Test	Specify the name of the test.	
Auto	As an alternative to typing in the name, click on this button to insert predefined auto-counter text. If you already have some text in the Test field, it is over-written by the auto-counter text.	Use Auto Naming and Auto Counters 
Status	Click on the drop-down arrow and select the current status of the test.	
Type	Click on the drop-down arrow and select the type of test.	
Run By	Click on the drop-down arrow and select the name of the person who ran the test.	
Checked By	Click on the drop-down arrow and select the name of the person who checked the test run.	
Last Run	Click on the drop-down arrow and select the date on which the test was last run.	
Description	Type a description of the test; you can format the text using the Notes toolbar at the top of the field. This text is also reflected in the Notes window, but cannot be edited there.	Notes Toolbar 
Input	Type in the input data.	
Acceptance Criteria	Type the acceptance or test success conditions.	
Results	Type the results of the last test.	

Topics

Topic	Detail	See also
Editing Scenario Tests	<p>To edit existing Scenario Test items for this element, either:</p> <ul style="list-style-type: none"> • Click on the item in the left-hand panel in Item mode • Double-click on the item in List mode to display the Test Details dialog, or • Click on the required item in the Testing folder in the Element Browser window (if this window is not displayed, click on the  icon in the Testing window toolbar); Scenario Test item icons have a Sc in the bottom right corner <p>To change the element for which to create test items, click on the required element in the Project Browser.</p>	<p>Test Details Dialog^[2214]</p> <p>Element Browser^[876]</p>

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

16.2.8 Move or Copy Tests Between Categories

When you define a test on the Unit, Integration, System, Acceptance or Scenario tab of the Testing window, you might decide that the test either is better suited to another category of tests, or forms a good template for tests in other categories. Enterprise Architect enables you to move or copy tests between categories.

How to

To move or copy a test

Step	Action	See also
1	<p>Open either:</p> <ul style="list-style-type: none"> • The Testing window, and the tab that contains the test you want to move or copy • The <i>Testing</i> folder of the Element Browser window 	<p>Element Browser^[876]</p>
2	<p>Right-click on the required test item in the list.</p> <p>The item context menu displays.</p>	
3	<p>Click on the appropriate option - Move to or Copy to.</p> <p>A list of test categories displays.</p>	

4	Click on the test category to which to move or copy the test. A confirmatory prompt displays.	
5	Click on the Yes button to confirm the move or copy.	
6	Click on the target tab of the Testing window to ensure that the test has been added, and make any required changes.	

Notes

- If you move or copy a test into the Scenario category, some unassociated data could be lost

Learning Center topics

- (Alt+F1) | **Testing | Testing | Changing Test Category**

16.2.9 Import Scenario as Test

You can import a scenario from a Use Case or other element, or from all elements in a package, into the Testing window Scenario tab. This avoids having to duplicate the scenario information manually.

The scenario description is copied to the Scenario Test Description tab in the Testing window. If a scenario contains a Structured Specification, its **Action** steps are also copied to the Description tab under the heading **Structured Specification**.

How to

To import a scenario from a single element

Step	Action	See also
1	Open a diagram or the Project Browser and select the element into which to import the scenario.	
2	Open either: <ul style="list-style-type: none"> • The Testing window and the Scenario tab • The <i>Testing</i> folder of the Element Browser window 	Element Browser ^[876]
3	Right-click on the list of tests to display the context menu, and select the Import element scenario(s) menu option. The Import Scenario dialog displays.	

Step	Action	See also
4	You can import scenarios from any element in the model by clicking on the Select element drop-down arrow and selecting the required element. Select the scenarios to import from the Select items to import list.	
5	Click on the OK button to import the selected scenario(s).	

To import scenarios from all elements in a package

Step	Action	See also
1	Open a diagram or the Project Browser and select the parent package element or an element within the package.	
2	Open either: <ul style="list-style-type: none"> The Testing window and the Scenario tab The <i>Testing</i> folder of the Element Browser window 	Element Browser ⁸⁷⁶
3	Right-click on the list of tests and select the Import Package Scenario(s) context menu option. The Import Scenario dialog displays. This version of the Import Scenario dialog lists all scenarios against all elements in the package; it does not enable you to select a specific element, but does enable you to filter the list of scenarios to those from specific types of element.	
4	In the Limit selection to these Object Types only field, type a comma-separated list of the object types for which to show scenarios. Click on the Refresh button.	
5	Click on the OK button to import the scenarios from each element as test scenarios for that element.	

Reference

The Import Scenario dialog has the following additional options:

Field	Usage	See also
Show related elements	Filter selection to apply only to related elements.	

Field	Usage	See also
only		
Limit selection to these Object Types only	Indicates specific element types, separated by commas, to filter for only those element types.	
Refresh	Refresh the list of available scenarios.	

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

Learning Center topics

- (Alt+F1) | **Execution Analysis | Testing | Import from Scenario**

16.2.10 Import Test From Other Elements

You can import any test from a Use Case or other element into the Testing window. This avoids having to duplicate the test information manually.

How to

To import a test

Step	Action	See also
1	Open a diagram or the Project Browser and select the element into which to import the tests.	
2	Open either: <ul style="list-style-type: none"> The Testing window, or The <i>Testing</i> folder of the Element Browser window 	Element Browser ⁸⁷⁶
3	Right-click on the list of tests and select the Import Tests from Other Element context menu option. The Import Element Tests dialog displays.	

Step	Action	See also
4	You can import tests from any element in the model by clicking on the Select element drop-down arrow and selecting the required element. Select the test to import from the Select items to import list.	
5	Click on the OK button to import the selected test(s).	

Reference

The Import Element Tests dialog has the following additional options:

Option	Action	See also
Show related elements only	Filter selection to apply only to related elements.	
Limit Selection to these Object Types only	Indicates specific element types, separated by commas, to filter for only those element types.	
Refresh	Refresh available options.	

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

Learning Center topics

- (Alt+F1) | [Execution Analysis](#) | [Testing](#) | [Import from Other Elements](#)

16.2.11 Import Responsibility or Constraint as Test

In the same way as you can import a scenario from an element as a test, you can also import an internal requirement (responsibility) or internal constraint as a test.

How to

To import a requirement or constraint as a test

Step	Action	See also
1	Open a diagram or the Project Browser and select the element into which to import the responsibility or constraint.	

Step	Action	See also
2	Open either: <ul style="list-style-type: none"> • The Testing window and the tab into which to import the test • The <i>Testing</i> folder of the Element Browser window 	Element Browser [876]
3	Right-click on the list of tests to display the context menu (if in the Element Browser, click on a test of the appropriate type).	
4	Click on the appropriate option, either: <ul style="list-style-type: none"> • Import element constraint(s) • Import element requirement(s) <p>The Import Constraint or Import Requirements dialog displays (the two dialogs are identical).</p> <p>The dialog lists all of the internal requirements or constraints in the selected element.</p>	
5	Select one or more of the items to import as tests, and click on the OK button. Each item is added to the list of tests in the Testing window, as a standard, 'Not Run' test.	
6	Edit the items to complete their definition as tests.	

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

Learning Center topics

- (Alt+F1) | **Execution Analysis | Testing | Import from Constraints**
- (Alt+F1) | **Execution Analysis | Testing | Import from Responsibility**

16.2.12 Create Maintenance Item From Test

If an element fails a test, one likely consequence is that a Defect (Issue) item has to be raised in model maintenance to correct the problem. You can generate this Defect item directly from the test that failed.

How to

To create a Maintenance item from a test

Step	Action	See also
1	In the Testing window, or the <i>Testing</i> folder of the Element Browser window, right-click on the test item from which to generate the Maintenance item.	
2	On the context menu, select the Create a Maintenance Defect from this test menu option. A message box displays.	
3	Click on the OK button to clear the message.	
4	Open the Maintenance window and select the Defects tab. The window displays a defect item having the same name as the test, and the test Description, Input, Acceptance Criteria and Results texts are displayed in the defect Description tab.	
5	Update the defect item as required.	Maintenance Item Properties <small>[2238]</small>

Notes

- You can create Maintenance Defect items from several Test items at once; press and hold (**Shift**) as you select the Test items, and then right-click and proceed as above - each selected Test item then generates a Defect item

Learning Center topics

- (Alt+F1) | **Testing | Testing | Create Maintenance Items**

16.2.13 Show Test Script Compartments

Any element that is capable of displaying a compartment can be used to show test scripts in a diagram. To make use of the feature the element must have an attached test.

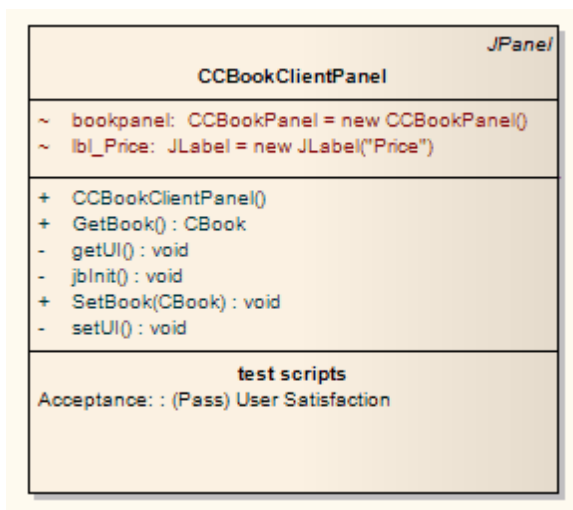
How to

To show the Test Script compartment on an element in a diagram

Step	Action	See also
1	Open a diagram containing the element with the attached test items.	
2	Double-click on the diagram background to display the Diagram Properties dialog.	

Step	Action	See also
	Click on the Elements tab.	
3	In the Show Compartments panel, select the Testing checkbox.	
4	Click on the OK button to save the setting. The tests now appear as an item in the test scripts compartment of the diagram element.	

Example



Learning Center topics

- (Alt+F1) | [Testing | Testing | Show on Daigrams](#)

16.2.14 Test Documentation

Enterprise Architect enables you to output the test scripts and results you have entered against elements in the model, in Rich Text format. For more information on entering test scripts and details see the previous sections of the Testing topic.

Access [Project Browser Package context menu | Documentation | Testing Report](#)

Notes

- You can also access the Generate Test Documentation dialog by selecting the **Project | Documentation | Testing Report** menu option
- The Generate Test Documentation dialog enables you to set up your report; you can configure which tests to include or exclude in the report, whether to include child packages and what file to output to

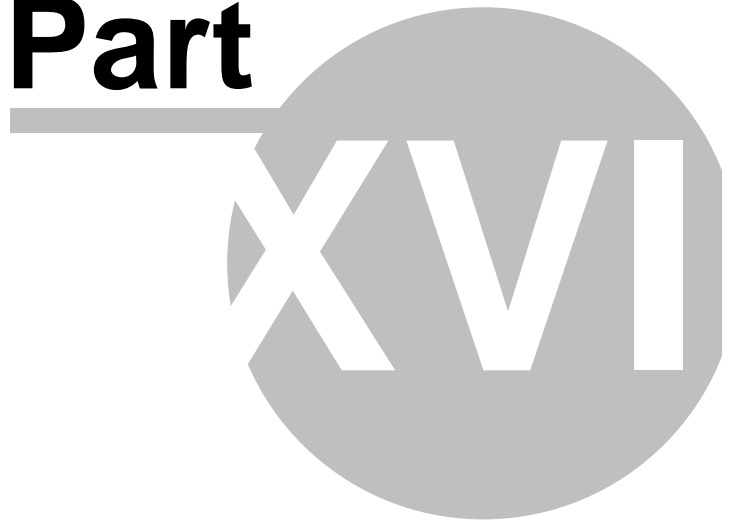
Learn more

- [Testing](#)^[2211]
- [Testing Report](#)^[2342]

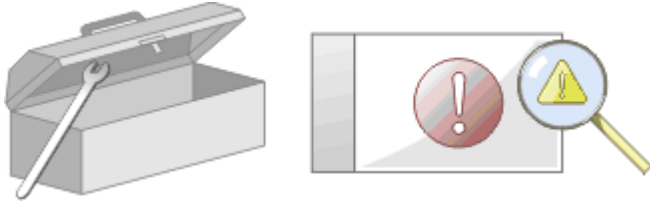
Learning Center topics

- (Alt+F1) | **Testing | Testing | Test Reports**

Part



17 Maintenance



This section discusses the Enterprise Architect facilities for:

- Creating and monitoring maintenance items on project elements
- Creating and monitoring change and issue items on the project

Learn more

- [Maintenance Items](#) ^[2236]
- [Changes and Defects](#) ^[2244]

Learning Center topics

- (Alt+F1) | **Maintenance**

17.1 Maintenance Diagram

A Maintenance diagram is a custom diagram used to describe change requests and issue items within a system model.







Change, Task and Issue elements can be linked back to other model elements in the system to illustrate how they must be modified, fixed or updated.

Maintenance models provide extensions to the UML model and enable change management of change items, and of the model elements that require the changes to be made to them.

Example Diagram [Example Maintenance Diagram](#) 

Tools

Select Maintenance diagram elements and connectors from the Maintenance pages of the Toolbox.

Maintenance Diagram Elements	Maintenance Diagram Connectors
 Package	 Aggregate
 Issue	
 Change	
 Test Case	
 Entity	

17.1.1 Example Maintenance Diagram

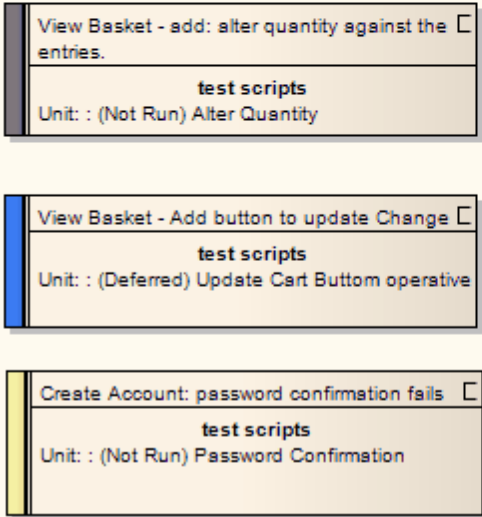
An example Maintenance diagram is shown below.

Changes

EA supports custom Elements of type 'Change'. These can be linked to other elements in the repository or used as a separate lists of any changes proposed for the model.

Below are a set of Change elements for the shopping basket with test definitions listed against them. They are ready for checking once they are confirmed as corrected.

The color markings reflect the Status of the element.



View Basket - add: alter quantity against the entries. <input type="checkbox"/>
test scripts
Unit: : (Not Run) Alter Quantity

View Basket - Add button to update Change <input type="checkbox"/>
test scripts
Unit: : (Deferred) Update Cart Button operative

Create Account: password confirmation fails <input type="checkbox"/>
test scripts
Unit: : (Not Run) Password Confirmation

17.2 Maintenance



Topics

Topic	Detail	See also
Maintenance Items	<p><i>Maintenance</i> items are defects, changes, issues and tasks. They all apply to individual model elements and can be used to record and capture problems, changes, issues and tasks as they arise, and document the solution and associated details. They are defined as follows:</p> <ul style="list-style-type: none"> • A defect can be considered as a failure to meet a requirement for the current model element • A change can be considered as a change in requirement for the current model element • An issue records a risk factor that might affect the project being recorded for the current model element • A task is a means of recording work in progress and work outstanding for the current model element <p>Note that each of these maintenance items applies at the model element level. For <i>changes</i>, <i>defects</i> and <i>issues</i> that apply to the whole system, see the Changes and Defects topic; for tasks that apply to the whole system, see the Project Tasks topic.</p>	<p>Changes and Defects ^[2243]</p> <p>Project Tasks ^[460]</p>
Creating and Editing Maintenance Items	<p>The following topics explain how to create and edit Maintenance items:</p> <ul style="list-style-type: none"> • <i>The Maintenance Workspace</i> - describes the Maintenance window • <i>Maintenance Item Properties</i> - describes how to complete the Maintenance window tabs for the various maintenance items • <i>Move or Copy Maintenance Items</i> - describes how to move items between maintenance categories or generate items from an item in a different category • <i>Create Elements From Maintenance Items</i> - describes how to generate elements from maintenance items 	<p>The Maintenance Workspace ^[2237]</p> <p>Maintenance Item Properties ^[2238]</p> <p>Move or Copy Maintenance Items ^[2240]</p> <p>Create Elements From Maintenance Items ^[2241]</p> <p>Insert Maintenance Feature ^[925]</p>

Topic	Detail	See also
	<ul style="list-style-type: none"> • <i>Show Maintenance Script in Diagram</i> - describes how to display maintenance items in elements on diagrams • <i>Insert Maintenance Feature</i> - describes how to add a maintenance item directly to an element via in-place editing 	

Learning Center topics

- (Alt+F1) | **Maintenance**


17.2.1 The Maintenance Workspace

Enterprise Architect makes it easy to record and capture problems and issues as they arise, and document the solution and associated details.

The Maintenance window provides a quick method of viewing and modifying the changes, issues, defects and 'to do' items associated with a particular model element. You can include the maintenance items in the main RTF documentation and HTML produced by Enterprise Architect. The RTF Setup dialog has checkboxes to show or hide element maintenance items.

Access [View](#) | **Other Element Tools** | **Maintenance (Alt+4)**

Topics

Topic	Detail	See also
Usage	<p>You can also use the Element Browser window to select and display specific items on the Maintenance window; click on the  icon in the Maintenance window toolbar to display the Element Browser, open the <i>Maintenance</i> folder and select the required item.</p> <p>In the folder, the 'page' icon contains a C for Change items, D for Defect items, T for Task items, or I for Issue items.</p> <p>The window has two formats; in:</p> <ul style="list-style-type: none"> • List mode a detailed list is provided showing several columns of information • Item mode this list is cut down and instead an editor for the selected item is provided <p>To toggle between the modes, click on the Show/Hide Properties button in the window toolbar:</p> <ul style="list-style-type: none"> • Item mode displays a single item with others of the same type listed in the left-hand panel; you can also 	Element Browser ⁸⁷⁶¹

Topic	Detail	See also
	<p>either switch to List mode or select another item from the Element Browser window</p> <ul style="list-style-type: none"> List mode displays all items of one type in the selected element; it does not, however, display as much detail on an item as Item mode does <p>Using the toolbar, you can add or delete items and show or hide the Properties window to enable you to edit each item in the list; click on the New icon in the window toolbar to add new items; in:</p> <ul style="list-style-type: none"> Item mode, this clears the fields for new data List mode, this displays the <item type> details for <element type> <element name> dialog <p>By clicking on the Auto button in Item mode or on the details dialog, you can apply an automatic naming/numbering nomenclature that you have previously defined. If you already have some text in the Name field, it is over-written by the auto-counter text.</p> <p>An asterisk on a tab indicates that the tab contains saved information; if the tab has no information or the information has not yet been saved, there is no asterisk.</p> <p>You can also display the maintenance items in a compartment of each appropriate element in a diagram.</p>	<p>Maintenance Item Properties ^[2238]</p> <p>Use Auto Naming and Auto Counters ^[799]</p> <p>Show Maintenance Scripts in Diagrams ^[2242]</p>

Learn more

- [Maintenance](#) ^[2236]

17.2.2 Maintenance Item Properties

Access [View](#) | [More Element Tools](#) | [Maintenance](#)

How to

To create, edit or delete maintenance items

Step	Action	See also
1	Open a diagram or the Project Browser and select an element.	

Step	Action	See also
	In Item mode, the oldest maintenance item of each type for that element is shown in the Maintenance window, on the appropriate tab. The other items are listed either in the left hand panel or in List mode.	
2	Click on the Browse Element icon in the window toolbar. The Element Browser displays.	
3	To: <ul style="list-style-type: none"> • Add a new item, select the appropriate tab, click on the New icon in the Maintenance window toolbar and complete the fields as described in the table below • Modify an existing item, select the item from the left-hand list panel or from the <i>Maintenance</i> folder of the Element Browser and edit the fields as described in the table below • Delete an existing item, select the item from the left-hand list panel or the Maintenance folder of the Element Browser window and click on the Delete icon in the Maintenance window toolbar 	
4	Click on the Save button in the window toolbar.	

Complete or edit the following fields on the Maintenance window

Field	Usage	See also
Name	Type the name or a short description of the defect.	
Auto	As an alternative to typing in the name, click on this button to insert predefined auto-counter text. If you already have some text in the Name field, it is over-written by the auto-counter text.	Use Auto Naming and Auto Counters 1799
Reported by	Select the name of the person who reported the defect.	
Reported	Select the date on which the defect was reported.	
Status	Select the defect status, such as New or Complete .	

Field	Usage	See also
Resolved by	Select the name of the person who fixed the defect.	
Resolved	Select the date on which the defect was resolved.	
Priority	Select the priority assigned to resolving the defect.	
Version	Type the version number associated with this fix.	
Description	Type a longer description of the defect; you can format the text using the Notes toolbar at the top of the field. This text is also reflected in the Notes window, but cannot be edited there.	Notes Toolbar ^[979]
History	Enter any notes or references to previous occurrences of this defect; you can format the text using the Notes toolbar at the top of the field. This text is also reflected in the Notes window, but cannot be edited there.	

Notes

- For information on element-level Defects, Issues, Changes and Tasks, see the *Maintenance* topic; for information on the Maintenance window, see the *Maintenance Workspace* topic
- The above table describes the fields of the Defects tab of the Maintenance window; the Changes, Issues and Tasks tabs differ only in minor details

Learn more

- [Maintenance](#)^[2236]
- [Maintenance Workspace](#)^[2237]

17.2.3 Move or Copy Maintenance Items

When you define an item on the Defects, Changes, Issues or Tasks tab of the Maintenance window, you might decide that the item either is better suited to another Maintenance category, or forms a good template for items in other categories.

Enterprise Architect enables you to move or copy items between categories.

How to

To move or copy a maintenance item

Step	Action	See also
1	Open the Maintenance window and select the tab that contains the item you want to move or copy.	Maintenance Workspace ^[2237]
2	Right-click on the required maintenance item. The item context menu displays.	
3	Click on the appropriate option - Move to or Copy to . A list of maintenance categories displays.	
4	Click on the category to which to move or copy the item. A confirmatory prompt displays.	
5	Click on the Yes button to confirm the move or copy.	
6	Click on the target tab to ensure that the item has been added, and make any required changes.	Maintenance Item Properties ^[2238]

17.2.4 Create Elements From Maintenance Item

A maintenance item identifies a defect, change, issue or task concerning an element. The maintenance item could itself be represented by an element if it has wider implications for the project or identifies - for example - an actor, activity or action that requires further definition.

You can create one or more elements from any maintenance item, using the Maintenance window. The new element is connected to the maintenance item's parent element by a Dependency connector. The item itself remains unchanged as a characteristic of its parent element.

Access [View](#) | [Other Element Tools](#) | [Maintenance](#) (Alt+4)

How to

To create an element from a maintenance item

Step	Action	See also
1	On the Maintenance window, select the tab that contains the item you want to create the element from.	Maintenance Workspace ^[2237]

Step	Action	See also
2	Right-click on the required maintenance item, and select the Create as new Element context menu option. The New Element dialog displays.	
3	Complete the fields on the dialog. Note that, in the Type field, you select the required element type; for example, you might create: <ul style="list-style-type: none"> • An Issue element for a Defect or Issue maintenance item • A Change element for a Change item, or • An Action for a Task item <p>You can, however, create a wide range of other element types should any of these be more appropriate, and use the Select Group button to select a profile, MDG Technology or Add-In to create an element specific to that element group.</p>	Add Elements Directly To a Package ^[798]

Learning Center topics

- (Alt+F1) | **Maintenance | Element from Item**

17.2.5 Show Maintenance Script in Diagram

Any element that is capable of displaying a compartment can show **maintenance scripts** in a diagram. To make use of the feature the element must have an attached maintenance item.

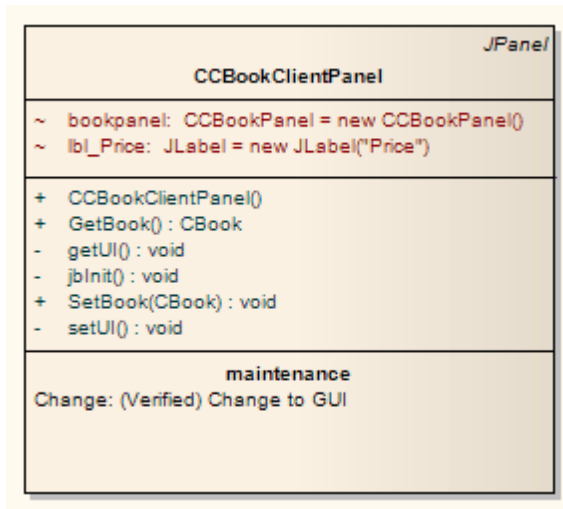
How to

To show maintenance compartments in a diagram

Step	Action	See also
1	Open a diagram containing the element with the attached maintenance items.	
2	Double-click on the diagram background to display the Diagram Properties dialog. Click on the Elements tab.	
3	In the Show Compartments panel, select the Maintenance checkbox.	
4	Click on the OK button to save the setting.	

The maintenance Items now appear as items in the maintenance scripts compartment of the diagram element.

Example



Learn more

- [The Maintenance Workspace](#) ²²³⁷

17.3 Changes and Defects

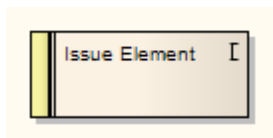


Topics

Topic	Detail	See also
Change and Defect Elements	<p>Changes and Defects are structured comments that can be used in managing change in a project.</p> <p>A Defect element (also known as an Issue element) corresponds to a failure to match the requirements for the current system.</p> <p>A Change element corresponds to a change in requirements for the current system.</p>	Changes ^[2246] Defects ^[2244]
Using Structured Comments	<p>You can track changes and defects (issues) in an Enterprise Architect model.</p> <p>Change and Defect elements can be created in UML diagrams and connected using Realization, Dependency, Aggregation and other relationships to show what model element each affects and how each is resolved.</p> <p>You can edit the element properties, and assign people (as Actor elements) to changes and defects.</p>	Element Properties ^[2247] Assign People to Defects or Changes ^[2247]

17.3.1 Defects (Issues)

A *Defect* (or *Issue*) element is a structured comment that contains information about defects and issues that relate to the system or model. This corresponds in some sense to a failure to meet defined requirements for the current system. An Issue element looks the same as a Requirement element:



Enterprise Architect enables you to generate and handle issues in much the same way as you can handle and **color code** Requirements. See the *Requirements* topic for more information.

You can link Issues using *Realize* connectors to model elements that are responsible for the defect. You can even structure a hierarchy of Issues using *Aggregation*.

How to

To add an Issue to the model using the Toolbox

Step	Action	See also
1	Open a Custom diagram.	
2	From the Custom pages or Common page of the Toolbox, drag the Issue icon onto the diagram.	Custom Group ^[716] Common Group ^[706]
3	Enter the details as required, on the Properties dialog.	Element Properties ^[2247]

To add an Issue to the model using the New Element dialog

Step	Action	See also
1	Right-click on a package in the Project Browser and select the Insert New Element context menu option. The New Element dialog displays.	
2	Complete the fields on the dialog - in the Type field, click on the drop-down arrow and select Issue .	Add Elements Directly To Packages ^[798]

Notes

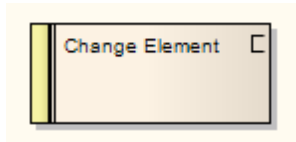
- Issue elements can be created with or without an identifying I in the top right corner of the element. To toggle the display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the Options dialog, Objects page

Learn more

- [Color Codes](#)^[1477]
- [Requirements](#)^[1468]
- [Objects](#)^[556]

17.3.2 Changes

A *Change* element is a structured comment that contains information about requested changes to the system/model. This corresponds in some sense to a change in requirements for the current system. A Change element looks the same as a Requirement element:



Enterprise Architect enables you to generate and handle Changes in much the same way as you can handle and color code Requirements. See the *Requirements* topic for more information.

You can connect *Changes* using *Realization* connectors to model elements that implement the Change, and you can structure a hierarchy of changes using *Aggregation*.

Change elements can be created with or without an identifying **C** in the top right corner of the element. To toggle the display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the Options dialog, Objects page.

How to

To add a Change to the model using the Toolbox, follow the steps below

Step	Action	See also
1	Open a Custom diagram.	
2	From the Custom pages or Common page of the Toolbox, drag the Change icon onto the diagram.	Custom Group ^[716] Common Group ^[706]
3	Enter the details as required, on the element Properties dialog.	Element Properties ^[2247]

To add a Change to the model using the New Element dialog, follow the steps below

Step	Action	See also
1	Right-click on a package in the Project Browser.	
2	Select the Insert New Element context menu option. The New Element dialog displays.	
3	Complete the fields on the dialog - in the Type field, click on the drop-down arrow and select Change .	Add Elements Directly To Packages ^[798]

Step	Action	See also

Learn more

- [Color Codes](#) ^[1477]
- [Requirements](#) ^[1468]
- [Objects](#) ^[556]

17.3.3 Element Properties

The Properties dialog for Changes and Issues is similar to that used by Requirements. It has a Properties tab containing the name of the Issue and relevant management details (such as owner and dates). You can also associate files with the issue and add Tagged Values.

Learn more

- [Requirement Properties](#) ^[1475]
- [Associate Files](#) ^[875]
- [Add Tagged Values](#) ^[971]

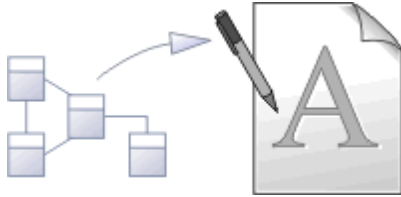
17.3.4 Assign People to Defects or Changes

As an example of how you might use the Relationship Matrix to monitor issues or changes, staff (actors) can be linked through *Realization* connectors to *Issues*. Each highlighted square indicates a responsibility of a staff member to work on or correct a named issue. This same approach can be used for any mix of model elements.

Part



18 Reporting



Model documentation is essential to realizing the full benefit of Enterprise Architect. Enterprise Architect provides a powerful mechanism for generating high quality, customized documentation directly from your model, in either RTF or HTML format.

Topics

Topic	Detail	See also
Specifying Content	<p>There are many ways to specify the Enterprise Architect content being documented; you can:</p> <ul style="list-style-type: none"> • Document a package and/or its child packages by manually highlighting the package and selecting a documentation control • Specify embedded packages for exclusion if child packages are recursively documented • Link a package to an RTF document template to simplify generating consistent types of documentation (for example, Use Case Reports) using the Documents feature • Select, group and order packages together in a different manner from the Project Browser by creating 'Virtual Documents', either linked through a master document with headers, footers and contents list, or as separate individual documents 	<p>Exclude Package From Report^[2260]</p> <p>Save As Document^[2325]</p> <p>Virtual Documents^[2327]</p>
RTF Documentation	<p>Rich text reports are documents produced by Enterprise Architect in Rich Text Format (RTF), which can be modified directly with RTF Style templates to alter the look and feel of generated output.</p> <p>Using MS Word you can further enhance the separate RTF documents output from the model by connecting and interweaving them into a linked master document with headers, footers and contents list.</p> <p>Enterprise Architect has a fully-featured RTF Document Generator that features:</p> <ul style="list-style-type: none"> • Powerful WYSIWYG RTF style template editor support • An easy-to-use document generator • An embedded RTF viewer that enables you to view RTF 	<p>RTF Documents^[2251]</p> <p>Use MS Word^[2344]</p>

Topic	Detail	See also
	documents generated by Enterprise Architect within Enterprise Architect	
RTF Reports	You can also generate a number of RTF reports on different aspects of your model.	Other Documents <small>2339</small>
HTML Documentation	Enterprise Architect provides automated web-based publishing of models, making it simple to explore large models efficiently on-line. Enterprise Architect enables the export of an entire model or a single branch of the model to HTML Web pages; you can also create web style templates to customize the HTML output.	HTML Reports <small>2352</small>

Learning Center topics

- (Alt+F1) | **Reporting**

18.1 RTF Documents



Topics

Topic	Detail	See also
Rich Text Format Documentation	<p>Rich text reports are documents produced by Enterprise Architect in Rich Text Format (RTF), a format common to many word processors.</p> <p>You can also maintain complex documents by creating virtual documents in Enterprise Architect, setting up a Master Document (package) element and/or Model Document elements (Class elements of stereotype Model Document) and linking packages into the document, in whatever order or combination is most appropriate to your requirements.</p> <p>You can select packages from different areas of the model, arrange them in any order, and edit or delete the packages; the virtual document automatically incorporates the changes each time you generate it.</p> <p>Alternatively, as RTF is particularly targeted at Microsoft Word™, it also enables you to link a number of rich text documents into a single master document.</p> <p>Typically you create a Word master document, then some Enterprise Architect RTF reports, and you link the reports back into sub-sections of the master document and refresh them as required during project development; in this way the project document becomes an easily-managed and feature-rich work product.</p> <p>You can also populate a Word document from specific sections of reports, based on bookmarks - for example, a Word document might have a section for a small part of your component model; using bookmarks you can generate a full component model, and then link into just one section of the report.</p> <p>This way you can maintain a complex Word document from parts of Enterprise Architect reports; the RTF Generator performs one pass for one template, but using a Word master document and Enterprise Architect bookmarks enables you to incorporate material from several RTF documents with different formats based on different templates.</p> <p>By adding tables of contents, figure tables, sections, and headers and footers, you can manage a complex document with relative ease; simply update the Enterprise Architect RTF reports then refresh the links in MS Word.</p>	<p>Virtual Documents [2327]</p> <p>Using MS Word [2344]</p> <p>RTF Bookmarks [2346]</p> <p>Diagram Tabs [695]</p>

Topic	Detail	See also
	<p>You can have several RTF reports open at the same time, as separate tabs in the central view area of the Enterprise Architect work area; you can also close the reports individually or all together, leaving views of other types (such as diagrams or code editors) still open.</p>	
The RTF Generator	<p>Enterprise Architect has an enhanced RTF Document Generator that features:</p> <ul style="list-style-type: none"> • Powerful WYSIWYG RTF style template editor support, enabling: <ul style="list-style-type: none"> • Headers and footers • Images • Indexes • Tabular Sections • Nested Sections • All model elements, connectors, diagrams and their properties • Template import and export using XML • Basic templates supplied for customization. • A document generator that: <ul style="list-style-type: none"> • Provides simplified options • Generates complex documents based on RTF templates • Has an embedded RTF viewer that you use to view RTF documents generated in Enterprise Architect directly within Enterprise Architect 	
More Information	<p>A tutorial on using the RTF Generator and creating RTF documentation is provided on the Sparx Systems website.</p>	<p>http://www.sparxsystems.com/resources/whitepapers/</p>

Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Generate Documents** permission to generate RTF documents

Learn more

- [Generate RTF Documents](#)  2253

- [Generate RTF Documentation Dialog](#) ^[2256]
- [RTF Document Options](#) ^[2261]
- [RTF Templates](#) ^[2270]
- [Custom Language Settings](#) ^[2326]
- [Include or Exclude a Package from Report](#) ^[2260]
- [List of Available Permissions](#) ^[264]

Learning Center topics

- (Alt+F1) | **Reporting | RTF Templates**
- (Alt+F1) | **Reporting | Virtual Documents**

18.1.1 Generate RTF Documents

Creating a Rich Text Format (RTF) document is a simple and flexible process. An RTF document is based on a package or an element in your project (more usually a package). To produce a document, you must select the package or element to report on in the Project Browser, Package Browser, Diagram List or Model Search.

You should also set the diagram properties to determine how the diagrams in the package are set out in the RTF document. When you have prepared and selected your package, use the context menu to open the Generate RTF Documentation dialog and configure the details of your document. The next topic guides you through creating a rich text report.

Topics

Topic	Detail	See also
Open the Generate RTF Documentation Dialog	<p>Use one of the following methods:</p> <ul style="list-style-type: none"> • Select the Project Documentation Rich Text Format (RTF) Report menu option • In the Project Browser, right-click on the required package and, on the context menu, select the Documentation Rich Text Format (RTF) Report menu option • In the Project Browser or a diagram, click on the required package or element and press (F8) • In a diagram, click on a specific element and select the Element Advanced Rich Text Format (RTF) Report menu option • In the Diagram List, Package Browser or Model Search, select one or more items, right-click and, from the context menu, select either the RTF Report Generate report for each selected object option or the RTF Report Generate one report for all selected option 	Generate RTF Documentation Dialog ^[2256]

Topic	Detail	See also
Generate RTF Report From Package Browser or Model Search	<p>When you select to create an RTF Report from the Diagram List, Package Browser or Model Search tools, you can generate an element-level report rather than a package-level report, and you have additional flexibility in selecting:</p> <ul style="list-style-type: none"> • The type of element to report on • The specific elements to report on, together or separately, whether in the same package or not <p>For example, you might want to find all elements with test cases, with the intention of reporting on some or possibly all such elements; with the Diagram List or Package Browser, you would identify these elements yourself within the list of all elements in a selected package, but with the Model Search you could specifically identify the elements across a section of the model or across the whole model, as required.</p> <p>The search filtering could be for specific test cases; however, the results are by element so if there are test cases outside the range in any element that has a filtered test, these elements are listed as well.</p> <p>Having generated the list of elements, you can select individual elements, blocks of elements, or all elements and then (as above) use the context menu to generate a report on all of the elements, or separate reports on each element.</p>	<p>Diagram List⁵⁹⁶</p> <p>Package Browser⁵⁸⁷</p> <p>Model Search⁶¹⁷</p>

How to

To generate and view an example RTF report right now

Step	Action	See also
1	Open the <i>EAExample</i> project.	
2	Open the <i>QA Model</i> package and right-click on the Testing package.	
3	Select the Documentation Rich Text Format (RTF) Report context menu option. The Generate RTF Documentation dialog displays.	
4	In the Output to file field, select a convenient file location in which to hold the generated report.	
5	In the Use Template field, click on the drop-down arrow and select (basic template).	

6	Click on the Generate button.	
7	When the report has been generated, click on the View button.	

Notes

- Reports can be configured to include all packages within a parent package, or just the top level

Learn more

- [Diagram Options](#) ^[2255]

18.1.1.1 Diagram Options

This topic refers to options on the Diagram page of the Diagram Properties dialog, which are used when generating RTF reports for a particular diagram.

Access [Diagram background context menu | Properties > Diagram page](#)

Topics

Topic	Detail	See also
Exclude Image from RTF Documents	Select this checkbox to exclude the image of the current diagram from any RTF reports.	
Document Each Contained Element in RTF	<p>Select this checkbox to ensure the RTF report generator includes details of elements that belong in other - external - packages but that are linked into this diagram within the package being reported on; the report includes external elements for each diagram on which you have selected the checkbox.</p> <p>On the Generate RTF Documentation dialog, the Include all Diagram Elements in Report checkbox defaults to selected, to include external elements for <i>every</i> diagram covered by the report; therefore, to include external elements only for specific diagrams, deselect this checkbox.</p> <p>In either case, you must enable the <i>Package.Diagram.Element</i> and <i>Package.Element</i> sections in your (customized) document generation templates in order to include this external element information in the report.</p>	<p>Generate RTF Documentation Dialog ^[2256]</p> <p>Report Elements From External Packages ^[2312]</p>
Divide Diagram Into Multiple Sections	<p>Select this checkbox to divide each large diagram into separate pages in the RTF document.</p> <p>This option is only effective when the Scaled Printing option is set to None on the Print Advanced dialog.</p>	Scaled Printing Option ^[766]

Topic	Detail	See also
Rotate Images	Select this checkbox to rotate each diagram image by 90 degrees in the RTF document; this is only valid for bitmap (.bmp) images.	

Learn more

- [Diagram Tab](#)^[727]

18.1.1.2 Generate RTF Documentation Dialog

The Generate RTF Documentation dialog enables you to set the exact contents and look and feel of your report.

Each tab of the dialog offers a number of RTF document generation options, as described in the following sections:

- Generate Tab (see below)
- RTF Templates
- Document Options
- Advanced Options
- Element Filters
- Other Filters
- Project Constants
- Word Substitution
- Language Substitution (Codepage)

When you have worked through these tabs and set the options you require, return to the Generate tab and click on the **Generate** button to produce your report.

Reference

Option	Action	See also
Model Document Root Element Root Package	Confirm the name of the element selected from the Project Browser, Diagram List or Model Search. If this is the specially-created model document element for a Virtual Document, the field is Model Document . Otherwise, this field identifies the selected element of the hierarchy to be reported on; that is, the Root Element or Root Package .	Virtual Document ^[2327]
Output to File	Type or select the location and filename for the generated documentation. The (...) (Browse) button enables you to navigate to the location.	

Option	Action	See also
Use Template	<p>Type or select the name of the RTF template to apply to document generation.</p> <p>You can select either a standard template (enclosed in parentheses) or a user-generated template.</p> <p>The standard templates include the following:</p> <ul style="list-style-type: none"> • (basic template) • (diagrams template) • (maintenance template) • (requirements template) 	
Output to PDF	Generate the report in PDF format rather than in RTF format.	
Use Language Substitutions	<p>Switch custom language word substitutions on.</p> <p>Deselect to switch custom language word substitutions off.</p>	Word substitutions [2268]
View Document On Completion	Open the document as soon as it has been generated.	
Use Internal Viewer	<p>Enable the View button to launch the generated RTF Documentation in the Enterprise Architect internal viewer.</p> <p>Deselect to enable the View button to launch the generated RTF Documentation in the MS Windows default RTF file viewer.</p>	Document options [2267]
Include all Diagram Elements in Report	<p>Include elements in the report from external packages that are referenced from a diagram, for every diagram covered by the report; this defaults to selected.</p> <p>The <i>Package.Diagram.Element</i> and <i>Package.Element</i> checkboxes must be selected in the current template.</p> <p>If external elements are to be included only for specific diagrams, deselect this checkbox and - only for the diagrams for which you want to include external elements in the report - for each diagram select the Document each contained element in RTF checkbox in the diagram properties.</p> <p>When both options are deselected, or when neither of the <i>Package.Diagram.Element</i> or <i>Package.Element</i> checkboxes are selected in the template, only elements in the current package are documented.</p>	Diagram properties [2265]
Optimize for Open Office	<p>Generate reports where diagrams are adjusted for clean rendering in Open Office.</p> <p>The option also overwrites the value fields defined in a</p>	

Option	Action	See also
	<p>document section with the actual value text.; if you open a generated report in Word, you can format the text but you cannot display the original field code.</p> <p>If the checkbox is <i>deselected</i>, diagrams displayed in Open Office are less distinct - value fields remain and are populated with the appropriate values; when you display a generated report in Word, you can right-click on the field and toggle between the value text and the field code.</p> <p>If you use Open Office as your text editor, you must select this checkbox to show the field values; Open Office defaults to showing the field codes only, and you cannot toggle to the field values.</p>	
Resource Document	Save the current options as a document definition.	Document definition ^[2259]
Edit Template	<p>Edit the currently-named template using the RTF Style Template Editor.</p> <p>You can only edit <i>user-defined</i> templates, not the standard templates provided with Enterprise Architect; standard template names are enclosed in parentheses.</p>	RTF Style Template Editor ^[2272]
Generate	Generate the document (after you have set all the options you require, on all tabs of the dialog).	
View	<p>Launch the generated RTF Documentation in:</p> <ul style="list-style-type: none"> • The MS Windows default RTF file viewer, or • The Enterprise Architect internal viewer, if you have selected the Use Internal Viewer checkbox 	
Abort	Cancel report generation; a prompt displays to confirm cancellation of the report.	

Notes

- For an introduction to generating RTF documentation, see the *RTF Documents* topic
- If you have never selected the **Optimize for Open Office** checkbox, the document generator prompts you to confirm which document style you want to adopt - Microsoft Office or Open Office; once you have made a selection in this prompt, the document generator always acts on the status of the checkbox and does not display the prompt again

Learn more

- [RTF Documents](#) ^[2257]
- [RTF Templates](#) ^[2270]
- [Document Options](#) ^[2261]
- [Element Filters](#) ^[2264]
- [Other Filters](#) ^[2267]
- [Project Constants](#) ^[2268]
- [Word Substitution](#) ^[2269]
- [Language Substitution \(Codepage\)](#) ^[2269]

18.1.1.3 Resource Documents

The *Resource Document* feature enables a particular documentation configuration to be 'remembered', linking the loaded template within the Generate RTF Documentation dialog to the current highlighted package. If a particular template is always used with a specific package, and multiple cases of documentation exist to be propagated, saving these as Resource Documents can ease document generation later.

How to

To create and use Resource Documents

Step	Action	See also
1	Open the Generate RTF Document dialog.	Generate RTF Documents ^[2253]
2	Click on the Resource Document button. The Save current as document definition dialog displays.	
3	In the Enter Value field, type a name for the document and click on the OK button. The document is added to the Resources window for easy future access.	
4	To generate documentation from the Resources window, right-click on the required document. The context menu displays.	
5	Select the required context menu option: <ul style="list-style-type: none"> • Open Document - Opens the corresponding .RTF file, as specified by the RTF template Filename property • Generate Document - Opens the Generate RTF Documentation dialog, loaded with the specified template 	

Step	Action	See also
	<ul style="list-style-type: none"> • Auto Generate Document - Generates documentation, with the document located at the path specified by the template's Filename property • Delete Document - Removes the specified document 	

Topic	Detail	See also
Batch Generate Resource Documents	<p>To generate a number of RTF documents at the same time, right-click on the <i>RTF Documents</i> folder name and select the Generate Documents context menu option; the Batch Document Generation dialog displays.</p> <p>The dialog lists all resource documents in the <i>RTF Documents</i> folder, defaulted to selected; deselect the checkbox against each document that you do not want to generate (or click on the Deselect All button to clear all selections, then select those you require).</p> <p>Click on the OK button to generate each of the remaining reports into their respective target file locations.</p> <p>The Generate All Documents context menu option automatically generates every document in the <i>RTF Documents</i> folder, without displaying the Batch Document Generation dialog.</p>	

Notes

- Resource Documents, saved into the Resources window, save the package, output file destination and template name from the Generate tab of the Generate RTF Documentation dialog, plus the data separately defined in the Options, Advanced and Element Filter tabs of the dialog

Learn more

- [Generate RTF Documentation Dialog](#)^[2256]
- [Document Options](#)^[2261]
- [Element Filter](#)^[2264]

18.1.1.4 Exclude Package from Report

You can exclude a particular package from any RTF or HTML report by marking it for exclusion.

Access **Project Browser package context menu | Documentation | Generated Report Options**

How to

To mark a package for exclusion from reporting

Step	Action	See also
1	Select the Generated Report Options menu option. The Generated Report Options dialog displays.	
2	Select the Exclude Package from Generated Reports radio button.	
3	Click on the OK button.	

Notes

- By default, packages are included in any generated reports
- When you exclude a package from reports, all of the selected package's subpackages are also excluded

18.1.1.5 Document Options

The RTF report options enable you to set type filters and order the elements.

Topics

Topic	Detail	See also
How to Access	<p>You can access the options from two different places; the start point affects the persistence of options selected:</p> <ul style="list-style-type: none"> • If you access the options on the Options and Advanced tabs of the Generate RTF Documentation dialog, you can define settings for the current report to be run; selections are non-persistent, and are reset when you exit the dialog or select a different template • If you access the options on the Document Options dialog, by clicking on the File Document Options menu option on the RTF Style Template Editor dialog, the settings are saved with the template as the default settings for any run of this report 	<p>Generate RTF Documentation ^[2256]</p> <p>File and Print Options ^[2277]</p> <p>RTF Style Template Editor ^[2272]</p> <p>RTF Element Filters ^[2264]</p> <p>Other ^[2267]</p>
Document Options dialog	<p>The Document Options dialog provides the options from both of the Options and Advanced tabs of the Generate RTF Documentation dialog, plus the Element Filters and Other Filters tabs, except for:</p> <ul style="list-style-type: none"> • The Switch generator button • The Disable large OLE file support option • The Insert page breaks when generating a Master 	<p>Generate RTF Documentation ^[2256]</p> <p>Element Filters ^[2264]</p> <p>Other Filters ^[2267]</p>

Topic	Detail	See also
	<p>Document option</p> <p>The dialog also contains the Optimize for Open Office option, as described for the Generate RTF Documentation dialog, and tabs for setting Element Filters and Other Filters.</p> <p>Click on the OK button to save your changes.</p>	

Reference

The Options tab of the Generate RTF Documentation dialog has the following fields:

Field	Usage	See also
Only include objects	<p>Filter elements according to date created or modified.</p> <p>In the first two fields, select the qualifiers from the drop-down lists; in the third field, select the appropriate date.</p>	
Where Package Phase	<p>Filter elements according to the value of the Package Phase field.</p> <p>In the first field select the qualifier, and in the second type the required phase (or leave the default value All).</p>	
With element status	<p>Filter elements according to status.</p> <p>In the first field, select the qualifier (like, not like, in, not in) and in the second field type the value to be used.</p> <p>Values should be enclosed in quotes; for example: "Proposed".</p> <p>If you type more than one value, separate them with a comma; for example: "Proposed", "Implemented".</p>	
Connector Direction	<p>Filter connectors according to direction.</p> <p>If you select Both, the connector is documented twice; once for the source element and once for the target.</p> <p>For the remaining two values, the connector is documented only for the source or target element, as appropriate.</p>	
Packages by	<p>Order packages in the generated documentation in either ascending or descending order of Name, Tree Order, Modified Date or Created Date.</p>	
Elements by	<p>Order elements in the generated documentation in either ascending or descending order of Name, Tree Order, Modified Date or Created Date.</p>	

Field	Usage	See also
Diagrams by	Order diagrams in the generated documentation in either ascending or descending order of Name, Tree Order, Modified Date or Created Date.	
Exclude details for	Exclude all elements of the selected type or types from the generated document.	
Exclude connector type	Exclude all connectors of the selected type or types from the generated document.	

The **Advanced** tab of the **Generate RTF Documentation** dialog has the following fields:

Field	Usage	See also
Hide 'note-less' elements	Exclude all elements without notes from the documentation.	
Diagram Format	Set the diagram format for the images included within the documentation to either Metafile or Bitmap.	
Skip root package	Exclude the parent package from the documentation and include only the child packages.	
No bookmarks	Stop RTF bookmarks being inserted into the generated document.	
Adjust Heading Levels	<p>Enable the RTF Generator to automatically restrict the levels of heading generated for nested sub-packages in a document</p> <p>The generator reproduces heading levels down to the value you set; for example, if you have four nested levels of sub-packages and you set this field to:</p> <ul style="list-style-type: none"> • Heading 2, all sub-packages in the report are documented under level 2 headings • Heading 4, the first level of subpackages are documented under level 2 headings, the next level under level 3 headings, and the remainder all under level 4 headings • Heading 6, the first level of subpackages are documented under level 2 headings, the next level under level 3 headings, the next under level 4 headings, and the next under level 5 headings; if you added further levels of sub-package they would all be documented under level 6 headings 	

Field	Usage	See also
	The field defaults to Heading 9 to accommodate the maximum number of levels of nested subpackages.	
Hide <Anonymous> elements	Hide anonymous elements in the documentation.	
Use style defined in template for notes	Override the character formatting specified in your Notes fields with the formatting set for the field in the template.	
Disable large OLE file support	Disable support for large Object Linking and Embedding (OLE) files.	
Insert page breaks when generating a Master Document	Insert a page break after each Model Document in a Master Document.	Model Document in a Master Document ^[2327]
Include child elements even if the parent element is filtered out	Document all child elements that are not otherwise filtered out by the restrictions, even if the parent elements are filtered out.	
Hide 'note-less' connectors	Exclude all connectors without notes from the documentation.	
Switch generator	Switch from this Generate RTF Documentation dialog (the Enhanced Template Driven Generator) to the Rich Text Format Report dialog (Legacy Generator). This button is not available if you displayed the dialog from the Diagram List, Package Browser or Model Search.	Rich Text Format Report dialog ^[2318]

18.1.1.6 Element Filters

The Element Filters tab enables you to include specific types of element in your report, rather than including every element type encountered.

Topics

Topic	Detail	See also
How to Access	<p>As with the Document Options tabs, you can access the Element filter details from two different places; the start point affects the persistence of the filter definition:</p> <ul style="list-style-type: none"> If you define the filters on the Element Filters tab of the Generate RTF Documentation dialog, you can create filter settings for the current report to be run; selections are non-persistent, and are reset when you close the dialog or select a different template If you access the filter definitions by clicking on the File Document Options menu option on the RTF Style Template Editor dialog, the settings are saved with the template as the default settings for any run of this report <p>You add filters by clicking on the Add Filters button; to edit the filters, either double-click on the panel contents or click on the Edit Filter button, to display the Edit Filters dialog.</p> <p>The format is the element field name, the conditions placed on the field value, any actual value or delimiting value to search on, and whether the filter item is required (mandatory), as described below.</p>	<p>Generate RTF Documentation ^[2256]</p> <p>File and Print Options ^[2277]</p> <p>RTF Style Template Editor ^[2272]</p> <p>Add Filters ^[627]</p>

Reference

Column/Button	Action	See also
Search In	Select the type and name of each element field to search on.	
Condition	<p>Select the condition of the search parameter.</p> <p>The available options are Contains, Equal To, Not Equals and One Of.</p>	<p>Fields and Conditions ^[629]</p>
Look For	<p>Specify the search term to perform the conditional search on.</p> <p>This value can pertain to the selected element field; for example, the value could be a date for <i>DateCreated</i> or a text value for other fields.</p> <p>The search term can contain multiple values, separated by commas.</p>	
Required	<p>Indicate that the search results must include elements with your search term in that field; you select these checkboxes on the Add Filters dialog.</p> <p>The fields listed as filters have an OR relationship when no Required checkboxes are selected; that is, if the search term is found in any one of those fields, then the element is displayed.</p> <p>Any field having the Required checkbox ticked overrides fields where</p>	<p>Adding Filters ^[627]</p>

Column/Button	Action	See also
	the Required checkbox is not ticked.	
Element Features - Optional - Required	<p>Specify whether element features are optional or required; these appear as a new branch underneath the root element term in the Search In column.</p> <p>If you scroll down the Search In column, you see sub-branches such as <i>Attribute</i>, <i>Change</i> and <i>Custom Property</i>; these are the element features.</p> <p>You can add these features by clicking on the Add Filter button; the Add Filters dialog displays, with a list of all the filters you can choose for an element or element feature.</p> <p>Click on the Search On drop-down arrow to see a list of the element features you can search on; each feature has its own set of filters such as <i>Name</i>, <i>Notes</i> and <i>Alias</i>, which you can add to your search.</p> <p>To search on an element attribute name, you would add the <i>Attribute</i> feature with a <i>Name</i> filter to your search.</p> <p>The Optional radio button enables you to generate a list of elements that meet <i>one</i> of the element features (<i>Element Type = Object</i>), or one of the feature filters (<i>Attribute Name = Class1</i>); for example, if your search is:</p> <p style="padding-left: 40px;"><i>Element Name = Class11, Attribute Name = m_Att1 or Scope = Public</i></p> <p>the search results list all the elements that have the <i>name</i> of <i>Class11</i> and all the elements that have an <i>Attribute Name</i> of <i>m_Att1</i> or a <i>Scope</i> of <i>Public</i>.</p> <p>The Required radio button enables you to generate a list of elements that <i>must</i> have the element features you have added; for example, if your search is:</p> <p style="padding-left: 40px;"><i>Element Name = Class, Attribute Name = m_Att1 or Scope = Public</i></p> <p>you would get elements that must have the <i>name</i> of <i>Class</i> AND an <i>Attribute</i> with a name of <i>m_att1</i> or a <i>Scope</i> of <i>Public</i>.</p>	
Add Filter	Add a new set of parameters to filter the search on.	Adding Filters [627]
Edit Filter	Open the Edit Filters dialog, which enables you to change the search parameters.	
Remove Filter	Remove the selected filter from the search.	

Learning Center topics

- (Alt+F1) | [Reporting | RTF Templates | Filter Elements](#)

18.1.1.7 Other Filters

The Other Filters tab enables you to define a set of filters to restrict your report to specific features of elements (sub-element components such as attributes, responsibilities or constraints). If the feature does not have the defined characteristics, it is not reported for the element.

Topics

Topic	Detail	See also
How to Access	<p>As with the Document Options tabs, you can access the filter details from two different places; the start point affects the persistence of the filter definition:</p> <ul style="list-style-type: none"> If you define the filters on the Other Filters tab of the Generate RTF Documentation dialog, you can create filter settings for the current report to be run; selections are non-persistent, and are reset when you close the dialog or select a different template If you access the filter definitions by clicking on the File Document Options menu option on the RTF Style Template Editor dialog, the settings are saved with the template as the default settings for any run of this report <p>You add filters by clicking on the Add Filters button; to edit the filters, either double-click on the panel contents or click on the Edit Filter button, to display the Edit Filters dialog.</p> <p>The format is the feature field name, the conditions placed on the field value, the actual value or delimiting value to search on, and whether the filter item is required (mandatory).</p> <p>The fields and options on the Other Filters tab are described below.</p>	<p>Generate RTF Documentation ^[2256]</p> <p>File and Print Options ^[2277]</p> <p>RTF Style Template Editor ^[2272]</p> <p>Add Filters ^[627]</p>

Reference

Column/Button	Action	See also
Search In	Select the name of each feature field to search on.	
Condition	Select the condition of the search parameter. The available options are Contains , Equal To , Not Equals and One Of .	Fields and Conditions ^[629]
Look For	Specify the search term to perform the conditional search on. This value can pertain to the selected field; for example, the value could be a date for <i>DateCreated</i> or a text value for other fields.	

Column/Button	Action	See also
	The search term can contain multiple values, separated by commas.	
Required	<p>Indicate that the search results must include elements with your search term in that field; you select these checkboxes on the Add Filters dialog.</p> <p>The fields listed as filters have an OR relationship when no Required checkboxes are selected; that is, if the search term is found in any one of those fields, then the element is displayed.</p> <p>Any field having the Required checkbox ticked overrides fields where the Required checkbox is not ticked.</p>	Adding Filters ⁶²⁷
Add Filter	Add a new set of parameters to filter the search on.	
Edit Filter	Open the Edit Filters dialog, which enables you to change the search parameters.	
Remove Filter	Remove the selected filter from the search.	

18.1.1.8 Project Constants

The Project Constants tab enables you to add values for your own project-specific fields in a report, to be inserted during compilation. The fields can be inserted into any template, in the text, headers or page headers and footers, using the RTF Style Template Editor context menu.

Field	Usage	See also
Add	<p>Add a new field and value to the list.</p> <p>A prompt displays for the two data items.</p>	
Delete	<p>Remove the selected field from the list.</p> <p>A prompt displays to confirm the deletion.</p>	
Import	<p>Import a set of fields from an external XML file.</p> <p>A browser displays to select the source file and location.</p>	
Export	<p>Export all fields to an external XML file.</p> <p>A browser displays to specify the target file and location.</p>	

Learn more

- [Add Content](#) ²³¹⁵

18.1.1.9 Word Substitution

The Word Substitution tab of the Generate RTF Documentation dialog enables you to define translations of technical terms used in Enterprise Architect, in particular field names, into a language other than English for direct substitution into RTF documentation.

How to

To add a translation for a term

Step	Action	See also
1	Double-click on the term in the English column in the Word Substitution list; the Enter Value field displays.	
2	Type the foreign language translation in the Enter Value field, and click on the OK button.	

18.1.1.10 Language Substitution

The Codepage tab of the Generate RTF Documentation dialog enables you to define languages other than English for direct substitution into RTF documentation.

If you export RTF-format documents from Enterprise Architect in languages other than English, you can customize the codepage, default language ID and character set that Enterprise Architect uses when generating RTF. This makes it much easier to generate documentation appropriate to your country or locale.

How to

Step	Action	See also
1	From the drop-down lists in the Language , Codepage and Charset fields, select the language, codepage and character set that most closely match your location.	
2	If required, modify the Substitute Tag by double-clicking on each and manually setting the value (for advanced use only).	
3	To clear the substitution list, double-click on each item in turn and delete the substitute value. Now when you generate RTF documents, the substitute tags are used in the output.	

Notes

- You can transport these tag definitions between models, using the **Export Reference Data** and **Import Reference Data** options

Learn more

- [Export Reference Data](#)^[306]
- [Import Reference Data](#)^[308]

18.1.2 RTF Templates

The Templates tab of the Generate RTF Documentation dialog enables you to create and delete your own RTF report templates. You can also select templates to edit, and import RTF templates saved as XML files.

Abstract

Topic	Detail	See also
	<p>In versions of Enterprise Architect later than 7.1, a <i>Normal.rtf</i> template is provided as a system template, as an external file stored in:</p> <p style="text-align: center;"><i>%APPDATA%\Sparx Systems\EA\RTF Templates</i></p> <p>This provides user-editable defaults of styles, numbering and other base formats; any styles modified in the Normal.rtf file reflect in newly created templates - one useful style is the default list numbering, <i>MasterList</i>.</p> <p>To edit the Normal.rtf template, use the Document Template Designer to create a new template called, for example, <i>Normal</i>, and use the File Import menu option to import the Normal.rtf file into the Normal template; modify the Normal template styles as required, but ensure there is no text when you save the template, just style definitions.</p> <p>Export the modified template (File Export) back into the Normal.rtf file in the <i>RTFTemplates</i> folder.</p> <p>A related feature in the Document Template Designer is the File Update Styles menu option, which enables you to update an existing template to reflect any changes to <i>Normal.rtf</i>.</p>	<p>File and Print Options^[2277]</p> <p>Stylesheets and Table of Contents^[2297]</p> <p>Update Styles^[2278]</p>

The Templates tab has the following functions

Field	Field	See also
Delete a template	Click on the template name and click on the Delete button.	
Create a new template	Click on the New button. The New Document Template dialog displays, on which you specify the template name and the name of any existing template to act as the base for the new	

Field	Field	See also
	<p>template.</p> <p>To make it easier to get up and running, Enterprise Architect provides a basic template with default settings on which you can base new templates.</p> <p>Modify the template as required, using the Document Template Designer.</p>	Document Template Designer ^[2272]
Edit	<p>Click on an existing template name and click on this button to make changes to the template.</p> <p>The template opens in the Document Template Designer.</p>	Document Template Designer ^[2272]

How to

To import RTF templates that were saved to batch XML files (using the **Project | Model Import/Export | Export Reference Data** menu option), follow the steps below:

Step	Action	See also
1	Click on the Import From Reference File button.	
2	On the Import Reference Data dialog, click on the Select File button and browse for and select the required file.	
3	<p>In the Select Datasets to Import panel, click on the required datasets. These can include one or more of:</p> <ul style="list-style-type: none"> • Templates - RTF Document • Templates - RTF Style Detail • Templates - RTF Tag & Language Options <p>Click on the Import button to import the templates. The names of the imported templates are added to the list on the Templates tab.</p>	Export Reference Data ^[308] Import Reference Data ^[308]

The Normal.rtf Template File

Notes

- If the template you want to import is a one-off copy (and not in a batch file) you can use the Document Template Designer **File | Import** menu option (and the **File | Export** option to export it to an XML file

again)

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is switched on, you must have **Configure Resources** access permission to create RTF templates
- In the Normal.rtf template, you must not edit the *SSbookmark* style; this defines the section styles and must be preserved

Learn more

- [Permission List](#)^[264]

Learning Center topics

- (Alt+F1) | **Reporting | RTF Templates**

18.1.2.1 Document Template Designer

Using the **Document Template Designer** you can create, edit and **test** your own **RTF document templates**, to customize the RTF documentation of your models in Enterprise Architect. You can then edit and develop the template file as required, and generate reports from it as you work to test the effects of your design.

Access Through the **Settings** main menu, the Generate Documentation dialog or the Project Resources window.

Settings | Document Template Designer

Generate Documentation Dialog > Templates | <template name>: Edit

Generate Documentation Dialog > General: Use Template <template name>: Edit Template, or

View | More Project Tools | Project Resources > Templates | Document Templates | Model |


Modify Document Template

Create a Template

- Creating a blank file to develop the template from scratch
- Copying one of the many useful templates provided with Enterprise Architect, including a simple 'base template'
- Copying an earlier customized template
- Importing an RTF template from an external file, such as the Normal.RTF style template provided with Enterprise Architect, or one of your own corporate files

The Document Template Designer

Facility	Usage	See also
Toolbar	The toolbar provides functions to create and save templates, to apply document options, and to test the output of the template.	

Facility	Usage	See also
	 <p>From left to right, the functions are</p> <ul style="list-style-type: none"> • Create a new template: select to enter the name of the new template and, if required, the name of an existing template to use as a base • Save: save changes to the current template • Save as: save the current template as a new template • Document Options: select to apply a range of options to the template that apply to all reports generated using the template (you can set equivalent options through the Generate RTF Dialog to apply only to the report currently being generated) • Select Template: <ul style="list-style-type: none"> • Displays the name of the template you have just created, or • Click on the drop-down arrow and select an existing template to view or edit <p>The template contents display in the Content panel</p> <p>System template names are enclosed in parentheses; you cannot edit these templates, but you can use the Save As option and edit the copy</p> • Filename: specify the filename of a test report generated from the template; type the file path and name, or click on the Select a File button to the right of the field and browse for the file location • Generate the Document - generate and display a report based on the template (before doing this, specify a target file and select a package, diagram or element in the Project Browser) • View the document - display the report in the file specified in the Filename field; you can return to a generated report many times whilst you review the effects of different sections of the template you are editing 	<p>RTF Templates ^[2270]</p> <p>Document Options ^[2261]</p>
Sections Panel	<p>Use the Sections panel to set the model components to be included in the report.</p> <p>The panel contains a hierarchy of model components and their</p>	

Facility	Usage	See also
	<p>properties, each with a checkbox that you select to include objects of that type in the report.</p> <p>As you select checkboxes in the Sections panel, corresponding pairs of open/close markers display in the Content panel, nested according to the position of the objects in the hierarchy.</p> <ul style="list-style-type: none"> Selecting the Package > Diagram > Connector checkboxes displays the following markers: <pre style="background-color: #ffffcc; padding: 5px;">package > diagram > connector > < connector < diagram < package</pre> <p>For detailed information on selecting and organizing the sections to include in the RTF report template, see the <i>Selecting Sections for Reporting</i> topic.</p>	<p>Selecting Sections for Reporting^[2304]</p>
<p>Content Panel</p>	<p>Use the Content panel to develop and edit the structure and content of the template and, therefore, the content and appearance of the reports generated from the template.</p> <p>As you add section markers to the template (as above), various pointer texts display between the them such as:</p> <ul style="list-style-type: none"> [right-click-to-insert-Element-field(s)] <p>Delete this text, right-click to display the context menu and select the Insert Field option to select from a range of appropriate data fields to include in the report. When the report is generated, the field is replaced by the actual value of the field</p> <ul style="list-style-type: none"> {Element.Name} is replaced by the name of the element <p>You can type or copy in any text you require around the field names, and use the context menu options (below) to format the text and insert, for example, tables, images, links and a contents list.</p> <p>The Content panel also has two text editing toolbars (based on the standard Windows toolbars) that you can use in conjunction with the context menu options.</p> <p>For further information on developing the content of the RTF template, see the <i>Add Section Content</i> topic.</p>	<p>Add Section Content^[2315]</p>
<p>Context Menu</p>	<p>When you right-click on the text within the Content panel, the context menu displays.</p> <p>The menu offers a wide range of options to apply styles and formatting to the template, and to save, import, and export the template file.</p> <p>For descriptions of these options, see the <i>Document Template</i></p>	<p>Document Template Designer Options^[2275]</p>

Facility	Usage	See also
	<i>Designer Options</i> section.	

Notes

- You can also transport RTF templates between models, using the **Export Reference Data** and **Import Reference Data** options

Learn more

- [RTF Templates](#) ^[2270]
- [Import RTF Template](#) ^[2316]
- [Generate RTF Documents](#) ^[2253]
- [Resources](#) ^[1003]
- [Export Reference Data](#) ^[306]
- [Import Reference Data](#) ^[308]

18.1.2.2 Document Template Designer Options

To access the Document Template Designer options, right-click in the body of the template to display the template context menu.

The following topics provide assistance on using the Document Template Designer.

Topic	Link
Scroll Through Text	Scroll Through Text ^[2276]
File and Print Options	File and Print Options ^[2277]
Cut and Paste Options	Cut and Paste Options ^[2279]
View Options	View Options ^[2281]
Image and Object Inserts	Image and Object Inserts ^[2282]
Character Formatting	Character Formatting ^[2284]
Paragraph Formatting	Paragraph Formatting ^[2285]

Topic	Link
Tab Support	Tab Support ^[2288]
Page Breaks and Repagination	Page Breaks and Repagination ^[2289]
Headers and Footers	Headers and Footers ^[2290]
Hyperlinks and Bookmarks	Hyperlinks and Bookmarks ^[2291]
Table Commands	Table Commands ^[2292]
Sections and Columns	Sections and Columns ^[2296]
Stylesheets and Table of Contents	Stylesheets and Table of Contents ^[2297]
User-Defined Section Numbering	User-Defined Section Numbering ^[2299]
Frame and Drawing Objects	Frame and Drawing Objects ^[2302]
Search/Replace Commands	Search/Replace Commands ^[2303]

Notes

- Throughout your template editing, be aware that:
 - To undo one or more immediately previous edits, press **(Ctrl+Z)**, or select the **Edit | Undo** menu option; you can still undo a change even after you have saved the change
 - To redo one or more immediately previous undone edits, press **(Ctrl+Y)**, or select the **Edit | Redo** menu option

18.1.2.2.1 Scroll Through Text

Scroll Using	Options
Keyboard Keys	Press: <ul style="list-style-type: none"> • (↑), (↓), (←) or (→) to scroll up or down a line, or left or right one character • (Home) to move to the beginning of the current line • (End) to move to the end of the current line • (Ctrl+Page Up) to move to the beginning of a file

Scroll Using	Options
	<ul style="list-style-type: none"> • (Ctrl+Page Down) to move to the end of a file • (Page Up) to display the previous page • (Page Down) to display the next page • (Ctrl+←) to move to the previous word • (Ctrl+→) to move to the next word • (Ctrl+↑) to move to the first column of the current line (if not already on the first column) or the first column of the previous line • (Ctrl+↓) to move to the first column of the next line • (F10) (or select the Other Jump menu option), type in a line number to jump to, and click on the OK button
Mouse	<p>Click on the vertical and horizontal scroll bar to perform various scrolling functions. These functions are available only if the horizontal or the vertical bar has been enabled by the startup parameters.</p> <p>Vertical Scroll Bar: Click on the arrows on either end to scroll the screen up or down by one line.</p> <p>Click above the elevator to scroll the screen up by one page; similarly, click below the elevator to scroll the screen down by one page.</p> <p>You can also drag the elevator to any position in the bar; as the elevator is dragged, the editor scrolls the screen up or down accordingly.</p> <p>Horizontal Scroll Bar: Click on the arrows on either end to scroll the screen left or right by one column; click on either side of the elevator to scroll the screen left or right by a 1/2 screen.</p> <p>You can also drag the elevator to any position in the bar; as the elevator is dragged, the editor scrolls the screen left or right accordingly.</p>

18.1.2.2.2 File and Print Options

Command	Action	Shortcut	See also
New	<p>Clear an existing template from the edit window and start an empty, unnamed template.</p> <p>The editor prompts you to save any modification to the previous template.</p>		
Revert	<p>Revert to the previously-saved copy of the template.</p>		
Save	<p>Save the text to the current file name.</p> <p>If a file is not yet specified, the editor prompts you for a template name.</p>	Ctrl + S	

Command	Action	Shortcut	See also
Save As	Similar to Save File , but you specify a new template name for saving the template.	Ctrl + Shift + S	
Import	Import an existing RTF document into the Template Editor, so as to insert model elements from that document into the template. This option is useful when creating templates from a predefined document with a particular 'look and feel'.		Import an existing RTF document ^[2316]
Export	Save the template as an RTF document rather than as a template. This can be useful for saving the template for other models.		
Update Styles	Imports the styles from <i>Normal.rtf</i> , found in the RTF Templates directory. This option is useful when creating a Master Document / Sub Documents that require consistent user-defined styles across multiple templates (including such things as numbering formats).		
Document Options	Display the Document Options dialog which enables you to set the filter and order the elements.		Document Options ^[2267]
Page Layout	Specify the page layout, before selecting the Print option. You can specify the margins (left, right, top and bottom) in inches.		
Printer Setup	Invoke a printer-specific dialog for the default printer (the default printer selection is made from the Windows Control panel). You select the parameters from a set of printer-specific options; these options include page size, page orientation, resolution and fonts.	Ctrl + Shift + P	
Print	Print the contents of the current file; the editor displays a dialog where you can select the scope of the printing. You can also choose to print only a selected part	Ctrl + P	

Command	Action	Shortcut	See also
	<p>of the file.</p> <p>To print a block of text, highlight the required text before invoking the Print function; this command prints a highlighted:</p> <ul style="list-style-type: none"> • Line block • Character block <p>The Print function prints on a default printer selected from the Windows Control panel; you can alter the printer setup or page layout prior to invoking the Print function.</p>		
Print Preview	<p>Preview the document before printing; the editor displays up to two pages at a time.</p> <p>You can scroll to a different page using (Page Up), (Page Down) or the scroll bar.</p> <p>By default the preview rectangle is sized to fit the current window; however, you can use the zoom option to enlarge or shrink the preview rectangle as required.</p> <p>Click on the Edit button or the File Print Preview menu option again to return to editing mode.</p>		
Close	<p>Close the Template Editor; the editor prompts you to save any unsaved information.</p>		

18.1.2.2.3 Cut and Paste Options

To	Action	See also
Highlight a word	Double-click on the word.	
Highlight a line	Move the cursor onto the line and press (F8).	
Select all file content	Press (Ctrl+A) or select the Edit Select All menu option.	
Copy a block	<p>Highlight the lines of text to be copied and press (Ctrl+C), or select the Edit Copy menu option.</p> <p>Move the cursor to the point at which to insert the text and press (Ctrl+V) or select the Edit Paste menu option.</p>	
Move a block	Highlight the lines of text to be moved and press (Ctrl+X), or select the Edit Cut menu option; the selected text is removed	

To	Action	See also
	<p>from the page.</p> <p>Move the cursor to the point at which to insert the text and press (Ctrl+V) or select the Edit Paste menu option.</p>	
Delete a block	Highlight the lines of text to be deleted and press (Delete).	
Delete a line	Press (Shift+F9) to delete the current line; the remaining lines close up.	
Paste special objects	<p>Select the Edit Paste Special menu option; the Paste Special dialog displays, listing the appropriate data type formats for pasting the copied object, as listed below:</p> <p>Click on the Paste button to embed the data into your application, or click on the Paste Link button to create a link to the original file.</p> <p><u>Native Object Format</u></p> <p>If available, this is the first format in the list box; you can edit data in this format using the original application, by double-clicking the object.</p> <p><u>Formatted Text</u></p> <p>A text format; this option offers the most suitable format if the data is pasted from another text output application, as the font and formatting attributes are reproduced accurately.</p> <p><u>Unformatted Text</u></p> <p>Another text format; this option pastes the text without retaining the formatting information.</p> <p><u>Picture Format</u></p> <p>The data is available in Picture format; you can later edit the object, by double-clicking on it and invoking the Microsoft MS Draw application.</p> <p>This format is preferred over the Bitmap and the Device Independent Bitmap formats.</p> <p><u>Device Independent Bitmap and Regular Bitmap formats</u></p> <p>The data is available in bitmap formats; you can later edit the object, by double-clicking on it and invoking the Microsoft MS Draw application.</p> <p>The editor converts these formats into the Picture format before calling the drawing application.</p>	

To	Action	See also


18.1.2.2.4 View Options

Menu Option	Action	See also
Page Mode	<p>Turn Page Mode on (the equivalent of <i>Print View</i> in Word) or off (the equivalent of <i>Normal View</i> in Word).</p> <p>In Page Mode, the editor displays one page at a time; it is most useful for documents containing multiple columns, as the columns are displayed side by side.</p>	
Fitted View	<p>Turn Fitted View on or off; this is a special case of Page Mode, in which the text wraps to the window width and the soft page breaks are not displayed.</p> <p>If you select Fitted View, Page Mode is automatically selected too; if you deselect Page Mode, Fitted View is automatically deselected too.</p>	
Ruler	<p>Display or hide the ruler at the top of the page.</p> <p>The ruler shows tab stops and paragraph indentation marks; it can also be used to create or delete tab stops.</p>	
Tool Bar	<p>Display or hide the tool bar above the ruler.</p> <p>The tool bar provides a convenient method of selecting fonts, point sizes, character styles and paragraph properties; the tool bar also shows the current selection for font, point size and character styles.</p>	
Status Ribbon	<p>Display or hide the status ribbon at the bottom of the editing panel.</p> <p>The status ribbon displays the current page number, line number, column number and row number; it also indicates the current insert/overtyping mode.</p>	
Paragraph Marker	<p>Display or hide the paragraph marker (an inverted 'P') at the end of each paragraph.</p> <p>This option is useful when working with lines with many different heights.</p>	
Hidden Text	<p>Show or hide 'hidden' text.</p> <p>Text formatted with the hidden attribute is shown with a dotted underline; when the option is turned off, the hidden text is not visible.</p>	<p>Character Formatting Options [2284]</p>

Menu Option	Action	See also
Field Names	<p>Insert, show and hide field names.</p> <p>As you develop your RTF document, you right-click on sections to insert field markers; you cannot insert these markers unless you have selected the Field Names option.</p> <p>When you deselect the option, existing field names are obscured.</p>	
Page Header/ Footer	<p>Display or hide the text of page headers and footers; if Page Mode is not selected, this option turns Page Mode on.</p> <p>When Page Mode is selected, you cannot edit the header or footer text unless you also select the Edit Edit Page Header/Footer menu option.</p> <p>When Page Mode is deselected, you can see and edit page headers and footers at the start of the document.</p>	Headers and Footers ^[2290]
Page Border	<p>Display or hide a page outline in Page Mode.</p> <p>When Page Mode is deselected, this option is not available.</p> <p>When Page Border is selected, the document contents are shown within a page outline; when Page Border is deselected, the document contents are formatted within the boundaries of the editing screen.</p>	
Zoom	<p>Shrink or enlarge the display of the document text, by selecting the appropriate percentage enlargement.</p> <p>The editor supports zoom percentages from 25 to 200.</p>	

18.1.2.2.5 Image and Object Inserts

To	Action	See also
Embed a picture in the document	<p>Position the cursor at the point at which to insert the picture bitmap or Windows metafile, and either:</p> <ul style="list-style-type: none"> • Select the Insert Embed Picture menu option, or • Press (Alt+F8) <p>A browser dialog displays, through which you select the picture to embed in the document.; the picture displays at the current cursor location.</p> <p>The embedded picture is saved within the document.</p>	
Link a picture file to the document	<p>Position the cursor at the point at which to link the picture bitmap or Windows metafile, and select the Insert Link Picture menu option.</p>	

To	Action	See also
	<p>A browser dialog displays, through which you select the picture to link to the document; the picture displays at the current cursor location.</p> <p>Linked picture data is not saved with the document, only the filename is stored within the document.</p>	
Embed an Ole object in the text	<p>Position the cursor at the point at which to embed the object, and select the Insert Ole Object menu option; the Insert Object dialog displays, listing the applications that are available to create the object.</p> <p>When you select an application, the editor launches it and you create the required object using this application; when you save the application, the editor inserts an icon or view that indicates the inserted object.</p> <p>You can later edit the object using the application, by double-clicking on the object.</p> <p>You can also use the Edit Paste Special menu option to import the OLE objects, provided that the object is available in the clipboard.</p>	Block editing 
Edit an embedded picture	<p>Click on the picture and select the Edit Edit Picture menu option; the Edit Current Picture Parameters dialog displays, through which you can change the width and height of the picture, in inches.</p> <p>You can also align the top, bottom, or middle of the picture with the base line of the text.</p>	
Edit an embedded Ole object	<p>Double-click on the icon that indicates the inserted Ole object; alternatively, position the cursor on the icon and select the Edit Edit Ole Object menu option.</p> <p>The editor opens the object in the application used to create it, and you can edit the object.</p>	
Insert a background picture for the text	<p>Select the Other Background Picture menu option; a browser dialog displays, through which you select the bitmap or metafile file to insert as a background picture.</p> <p>The picture occupies the entire text area.</p> <p>To remove the background picture, deselect the Background Picture menu option.</p>	
Insert an RTF File	<p>Position the cursor at the point at which to insert the file, and select the Insert Insert RTF File menu option.</p>	

18.1.2.2.6 Character Formatting

When you change the format of existing text, any new characters you type immediately following automatically assume the formatting characteristics of the existing text.

To	Action	See also
Apply character format	<p>Highlight the text to which to apply the format, and use one or more of the following menu options or key combinations, as required:</p> <ul style="list-style-type: none"> • Font Normal, or press (Alt+0) • Font Bold, or press (Ctrl+B) • Font Underline, or press (Ctrl+U) • Font Double Underline, or press (Ctrl+D) • Font Italic, or press (Ctrl+I) • Font Superscript, or press (Alt+4) • Font Subscript, or press (Alt+5) • Font Strike, or press (Alt+6) (puts a line through the text) • Font All Caps • Font Small Caps <p>To reset any character format, highlight the text and select the Font Normal menu option, or press (Alt+0).</p>	
Change font typeface and point size	<p>Highlight the text to change and select the Font Fonts menu option, or press (Alt+F10).</p> <p>The Font Selection dialog displays, from which you select the required typeface and point size.</p> <p>Click on the OK button.</p>	
Change character style	<p>Highlight the text to change and select the Font Style menu option.</p> <p>The Select a Style dialog displays, listing the currently-defined character styles in the template stylesheet.</p> <p>Select the required style and click on the OK button.</p>	Template stylesheet ²²⁹⁷
Change the color of text, background (permanent highlight) or underline	<p>Highlight the text to change and select one or more of the following options, as required:</p> <ul style="list-style-type: none"> • Font Text Color • Font Background Color • Font Underline Color <p>In each case, the Color dialog displays, through which you can select or define the required color.</p> <p>When you have selected a color, click on the OK button.</p>	

To	Action	See also
Change character spacing	<p>Normal character spacing is 20 Twips; if you want to change this (or return to it), highlight the text to adjust, and select the Font Spacing menu option.</p> <p>The Character Spacing dialog displays.</p> <p>Select the radio button to expand or compress spacing, or to return to normal spacing; if you are changing from normal spacing, enter the number of Twips to set the spacing to.</p> <p>Click on the OK button.</p>	
Hide text	<p>Hidden text is not displayed on the screen or printer, but remains in the document and is not deleted.</p> <p>Highlight the text to hide and select the Font Hidden menu option, or press (Ctrl+H).</p> <p>The highlighted text is not displayed and the rest of the text closes up.</p> <p>To view hidden text, select the View Hidden Text menu option; you can then make the text normal again by highlighting it and deselecting the Font Hidden menu option.</p>	
Box text	<p>Highlight the text to box and select the Font Boxed menu option; this creates a broken-line border around the selected text.</p>	
Insert a non-breaking space	<p>Move the cursor to the point at which to insert the non-breaking space and select the Insert Non-breaking Space menu option.</p>	
Insert a non-breaking dash	<p>Move the cursor to the point at which to insert the non-breaking dash and select the Insert Non-breaking Dash menu option.</p>	
Insert an optional hyphen	<p>Move the cursor to the point at which to insert the hyphen and select the Insert Optional Hyphen menu option.</p>	

18.1.2.2.7 Paragraph Formatting

The functions described below operate on the current paragraph, or on a **highlighted** block of text.

To	Action	See also
Clear all paragraph formatting	<p>Select the Paragraph Normal menu option.</p> <p>The editor pushes the paragraph back up to the page margin.</p>	

To	Action	See also
Set text flow in document	<p>To set the text flow for:</p> <ul style="list-style-type: none"> • A selected block of text, select the Paragraph Text Flow menu option; the Paragraph Text Flow dialog displays • The entire document, select the Edit Document Text Flow menu option; the Document Text Flow dialog displays <p>In either case, select the required text flow direction and click on the OK button.</p>	
Center text	Select the Paragraph Center menu option or press (Alt+8).	
Right-justify text	Select the Paragraph Right Justify menu option or press (Alt+9).	
Justify both sides of text	Select the Paragraph Justify Both menu option.	
Set double line spacing	<p>Select the Paragraph Double Space menu option.</p> <p>A double-spaced paragraph has a blank line after each text line.</p>	
Indent paragraph left	<p>Select the Paragraph Indent Left menu option or press (Alt+L).</p> <p>Select the option again to increase the indent.</p>	
Indent paragraph right	<p>Select the Paragraph Indent Right menu option or press (Alt+R).</p> <p>Select the option again to increase the indent.</p>	
Create hanging indent	<p>Select the Paragraph Hanging Indent menu option or press (Alt+T).</p> <p>Select the option again to increase the indent of all lines below the first.</p>	
Keep paragraph lines together	<p>Select the Paragraph Keep Together menu option.</p> <p>The editor attempts to keep all lines within the paragraph on the same page.</p>	

To	Action	See also
Keep paragraph with next	Select the Paragraph Keep with Next menu option The editor attempts to keep the last line of the current paragraph and the first line of the next paragraph on the same page	
Prevent 'widow' and 'orphan' lines	Select the Paragraph Widow/Orphan Control menu option. The editor attempts to avoid having: <ul style="list-style-type: none"> • the first line of the paragraph on the previous page ('widow' line) • the last line of the paragraph on the next page ('orphan' line) 	
Start text on new page	Move the cursor to the point at which to start the new page, and select the Paragraph Page Break Before menu option.	
Insert border and shading for text block	Highlight the required text and select the Paragraph Border and Shading menu option The Paragraph Box Parameters dialog displays, on which you specify: <ul style="list-style-type: none"> • which sides of the box to display (including a line between text lines) • whether the lines are thick or doubled • the degree of gray shading behind the text • the color of the lines 	
Define line spacing	Highlight the required lines and select the Paragraph Paragraph Spacing menu option. The Paragraph Spacing Parameters dialog displays, on which you specify the line spacing and the point spacing before and after lines.	
Set a background color for text space	Highlight a text string or block of text and select the Paragraph Background Color menu option. The Color dialog displays, on which you select the background color. The editor highlights the full width of the page in that color, for the selected lines.	
Create a bulleted list	Highlight the required lines of text and select the Paragraph Bullet menu option. The editor formats the lines into a simple bullet list.	

To	Action	See also
Create a numbered list	Highlight the required lines of text and select the Paragraph Numbering menu option. The editor formats the lines into a simple numbered list.	
Apply numbering to paragraphs	1) Set up a numbering list and overrides (Edit List and Overrides). 2) Apply the numbering levels to the template sections (Paragraph List Numbering).	User-Defined Section Numbering ^[2295]
Apply a paragraph style from the template stylesheet	Select the Paragraph Style menu option. The Select a Style dialog displays, listing the currently-defined paragraph styles in the template stylesheet. Select the required style and click on the OK button.	Stylesheets and Table of Contents ^[2297]

Learn more

- [Block Editing](#) ^[2279]

18.1.2.2.8 Tab Support

The RTF Style Template Editor supports *left*, *right*, *center* and *decimal* tabs. Tabs are very useful for creating columns and tables. A paragraph can have as many as twenty tab positions.

A tab usually applies to every line of the current paragraph. However, if you highlight a block of text before setting a tab, the tab then applies to every line in the highlighted text.

Topics

Topic	Detail	See also
Create and Set Tabs	You can create tabs quickly and easily using the ruler at the top of the screen To create a: <ul style="list-style-type: none"> • Left tab, click on the required tab position on the ruler; the left tab is indicated on the ruler by an L shape • Right tab, right-click on the required tab position on the ruler; the right tab is indicated on the ruler by a reversed L shape • Center tab, press (Shift) and click on the required tab position on the ruler; the center tab is indicated on the ruler by an inverted T shape 	

Topic	Detail	See also
	<ul style="list-style-type: none"> • Decimal tab, press (Shift) and right-click on the required tab position on the ruler; the decimal tab stop is indicated on the ruler by an inverted T shape with a dot on the right hand side <p>This tab is for numbers with a decimal point; numbers scroll left from the tab until you type a point, then numbers scroll right.</p> <p>You can also set tabs using the Paragraph Set Tab menu option, which displays the Set a Tab Position dialog; this enables you to specify the tab type, and provides two advantages over the ruler: you can set the tab position with more precision and with a clear value that you can duplicate; and you can add a tab leader line (dot, hyphen, or underline)</p> <ul style="list-style-type: none"> • To clear a single tab position for selected text, select the Paragraph Clear Tab menu option; the Clear a Tab Position dialog displays, on which you select the tab to clear • To clear all tab stops for selected text, select the Paragraph Clear All Tabs menu option • To move a tab position using the mouse, click on the tab symbol on the ruler and drag it to the new position 	

Notes

- The **Other | Snap To Grid** menu option affects the movement of the tabs (and the paragraph indentation markers) on the ruler; when you select this option, the movements of the tab markers are locked on to an invisible grid at intervals of 1/16 inch (half a ruler division)

18.1.2.2.9 Page Breaks and Repagination

You can force a page break in the document by selecting the **Insert | Insert Break | Page Break** menu option, or by pressing **(Ctrl+Enter)**. The forced page break is indicated by a solid line in the editing window.

If **Page Mode** is off, the editor also marks automatic page breaks when the text overflows a page; these are indicated by a dotted line.

You can repaginate your document, using the **Edit | Repaginate** menu option. This updates the **Page Number** and **Page Count** fields, and recompiles the table of contents.

You insert the **Page Number** and **Page Count** fields as follows:

Topic	Detail	See also
Insert the page number	Position the cursor at the point at which to display the page number, and select the Insert Page Number menu option.	

Topic	Detail	See also
	The page number is displayed in gray.	
Insert the page count	Position the cursor at the point at which to display the total number of pages in the document, and select the Insert Page Count menu option. The page count is displayed in gray.	

18.1.2.2.10 Headers and Footers

To	Action	See also
Edit the page header and footer text	In Page Mode , select the Edit Edit Page Header/Footer menu option. A paragraph marker displays at the top and bottom of each page, and you can type in, format or delete the appropriate text. If Page Mode is turned off, all page headers and footers are displayed in a block at the start of the document, with identifying labels; you can also edit the text here. Each section in a document can have its own page header and footer.	
Create the header and footer for the initial page of the document	In Page Mode , select the Edit Edit Page Header/Footer menu option and then select the Edit First Page Header/Footer Create First Page Header or Create First Page Footer menu option. A paragraph marker displays at the top or bottom of the first page, and you can type in and format the appropriate text. If Page Mode is turned off, all page headers and footers are listed in a block at the start of the document, with identifying labels; you can also edit the text here.	
Delete the header and footer for the initial page of the document	Whilst you can delete the first page header or footer text by simple editing, you must specifically delete the 'first page' assignment in order to display the header and footer text of the next section on the first page. In Page Mode , select the Edit Edit Page Header/Footer menu option and then select the Edit First Page Header/Footer Delete First Page Header or Delete First Page Footer menu option. This removes the first page text and assignment, and displays the next-defined header and footer text.	
Create a footnote	Move the cursor to the position at which to insert the footnote marker, and select the Insert Footnote/Endnote Footnote menu option. The Footnote Parameters dialog displays, on which you enter the footnote marker and footnote text, and select whether to make the marker a superscript. Click on the OK button; the editor inserts the footnote marker at the	

To	Action	See also
	current cursor location and, in Page Mode , displays the footnote text at the bottom of the page.	
Edit footnote text	<p>Select the Edit Edit Footnote/Endnote Edit Footnote Text menu option; the text of each footnote displays in the document text where its marker was inserted.</p> <p>Locate the text and make the required changes; in Page Mode the modified footnote displays at the bottom of the page.</p> <p>When you have finished editing footnote text, <i>deselect</i> the Edit Edit Footnote/Endnote Edit Footnote Text menu option; the footnote text is no longer shown in the document text.</p>	
Create an endnote	<p>Move the cursor to the position at which to insert the endnote marker, and select the Insert Footnote/Endnote Endnote menu option.</p> <p>The Endnote Parameters dialog displays, on which you enter the endnote marker and endnote text, and select whether to make the marker a superscript.</p> <p>Click on the OK button; the editor inserts the endnote marker at the current cursor location and, in Page Mode, displays the endnote text at the end of the section or document.</p>	
Edit endnote text	<p>Select the Edit Edit Footnote/Endnote Edit Endnote Text menu option; the text of each endnote displays in the document text where its marker was inserted.</p> <p>Locate the text and make the required changes; in Page Mode, the modified endnote displays at the bottom of the page.</p> <p>When you have finished editing endnote text, <i>deselect</i> the Edit Edit Footnote/Endnote Edit Endnote Text menu option; the endnote text is no longer shown in the document text.</p>	

18.1.2.2.11 Hyperlinks and Bookmarks

To	Detail	See also
Manage bookmarks	<p>Each template contains a number of bookmarks that mark the sections; you can apply these bookmarks to related sections, create and assign your own, delete those that are not required, and locate specific bookmarks in the document.</p> <p>In Page Mode, select the Insert Bookmark menu option; the Bookmark dialog displays</p> <ul style="list-style-type: none"> To assign a bookmark to the current cursor position, either type a new bookmark in the top field or select one from the list, and click on the Insert button To delete an existing bookmark, click on it in the list and click on 	

To	Detail	See also
	<p>the Delete button</p> <ul style="list-style-type: none"> To mark a bookmark with the number of the page it is on, click on the bookmark in the list and click on the Set Page Reference button To locate a bookmark in the text, click on it in the list and click on the Go to button 	
Insert hyperlinks	<p>Right-click on the point at which to create the hyperlink and select the Insert Hyperlink context menu option; the Insert Hyperlink dialog displays.</p> <p>You can create a hyperlink within an RTF document to an external document, Help topic or web page.</p> <p>In the Link Text field, type the text to be hyperlinked and, in the Link Code field, type or paste the web page URL, help topic file or external file path and name.</p> <p>To capture the help topic file name, right-click on the displayed topic, select the Properties context menu option, and copy the file name - when you insert the file name in the Link Code field, ensure that the file name has the prefix <code>\$Help://</code>.</p> <p>Click on the OK button; the hyperlinked text displays in the document.</p> <p>Double-click on the link to display the web page or external document.</p>	

18.1.2.2.12 Table Commands

The **Table** menu enables you to create a new table, or to edit an existing table's attributes.

To	Action	See also
Insert a table in the document	<p>Position the cursor at the appropriate point, and select the Table Insert Table menu option; the New Table Parameters dialog displays, in which you specify the number of table rows and columns.</p> <p>The editor initially creates cells of equal width; you can, however, change the cell width by dragging the cell borders using the mouse.</p> <p>When Page Mode is deselected, the table structure is not visible.</p>	
Add a header row	<p>Select the rows to act on, and then select the Table Header Row menu option.</p> <p>Apply any heading settings and formatting to the rows in the highlighted block.</p>	
Insert a new row above the current row	Select the Table Insert Row menu option.	

To	Action	See also
Insert a new column to the left of the current column	Select the Table Insert Column menu option.	
Merge cells	Select the cells to merge and select the Table Merge Cells menu option. The width of the resulting cell is equal to the sum of the merged cells. You can merge cells across a row, down a column, and in a block spanning both rows and columns.	
Split a cell	Select the cell to split and select the Table Split Cell menu option. The selected cell is split into two cells of equal width; any text in the original cell is assigned to the first cell, and the second cell is created empty.	
Delete cells	Select the cells to delete and select the Table Delete Cells menu option; the Delete Table Cells dialog displays, on which you specify whether to delete: <ul style="list-style-type: none"> • Cells - deletes the highlighted cells • Columns - deletes all the cells in the highlighted column or columns • Rows - deletes all the cells in the highlighted row or rows If you delete all cells in a table, the table itself is automatically deleted.	
Position the table on the page	Click on any part of the table and select the Table Row Position menu option; the Table Row Alignment dialog displays, on which you select to left-align, center or right-align the table on the page. This option has little effect if the table is wide enough to span the page or text column.	
Set the height of a row, or all rows	Select the row to adjust and select the Table Row Height menu option. The Row Height Parameters dialog displays, enabling you to set an automatic row height, a minimum row height, or an exact row height. You can apply the setting to the selected rows only, or to all rows in the table.	
Keep row text together if it continues over a page	Select the rows to protect (preferably all rows in the table) and select the Table Keep Row Together menu option. If the row continues over the end of the page, the whole row is moved to the top of the next page.	

To	Action	See also
Set text flow in rows	<p>Select the rows and select the Table Row Text Flow menu option; the Table Text Flow dialog displays, on which you select the direction of flow of the text and select to apply the setting to the selected rows or all rows in the table.</p> <p>This option also moves the whole row over to the appropriate side of the page or column.</p>	
Set the width of selected cells	<p>Select the cells to act on and select the Table Cell Width menu option; the Set Cell Width dialog displays, on which you set the cell width and text margin and apply them to:</p> <ul style="list-style-type: none"> • All cells in a highlighted block • The selected cells only • All cells in the selected column or columns, or • All cells in the selected row or rows. 	
Define the cell border width	<p>Select the cells to act on and select the Table Cell Border Width menu option; the Set Cell Border dialog displays, on which you set the width of the lines at any or all of the top, bottom, left and right of a cell, or whether to draw a uniform border around the cells.</p> <p>You can also set the text margin, and apply all the settings to:</p> <ul style="list-style-type: none"> • All cells in a highlighted block • The selected cells only • All cells in the selected column or columns, or • All cells in the selected row or rows. 	
Define the cell border color	<p>Select the cells to act on and select the Table Cell Border Color menu option</p> <p>The Set Cell Border Color dialog displays, on which you set the color of the lines at any or all of the top, bottom, left and right of a cell, or whether to have a uniformly colored border around the cells.</p> <p>You then apply the settings to:</p> <ul style="list-style-type: none"> • All cells in a highlighted block • The selected cells only • All cells in the selected column or columns, or • All cells in the selected row or rows 	
Define the cell shading	<p>Select the cells to act on and select the Table Cell Shading menu option; the Cell Shading Parameters dialog displays, on which you set the shading percentage.</p> <p>The value 0 indicates the palest background, whereas the value 100</p>	

To	Action	See also
	<p>indicates a black background; you apply the setting to:</p> <ul style="list-style-type: none"> • All cells in a highlighted block • The selected cells only • All cells in the selected column or columns, or • All cells in the selected row or rows 	
Define the cell background color	<p>Select the cells to act on and select the Table Cell Color menu option; the Cell Color Parameters dialog displays, on which you set the cell background color.</p> <p>You then apply the color to:</p> <ul style="list-style-type: none"> • All cells in a highlighted block • The selected cells only • All cells in the selected column or columns, or • All cells in the selected row or rows 	
Vertically align cells	<p>Select the cells to act on and select the Table Cell Vertical Align menu option; the Cell Vertical Alignment dialog displays, on which you select to align the selected cells by top, center, bottom or baseline.</p> <p>You then select to align:</p> <ul style="list-style-type: none"> • All cells in a highlighted block • The selected cells only • All cells in the selected column or columns, or • All cells in the selected row or rows 	
Rotate cell text	<p>Select the cells to act on and select the Table Cell Rotate Text menu option; the Cell Text Rotation dialog displays, on which you select to display text horizontally across the cell, vertically up the cell, or vertically down the cell.</p> <p>You then select to apply the rotation to:</p> <ul style="list-style-type: none"> • All cells in a highlighted block • The selected cells only • All cells in the selected column or columns, or • All cells in the selected row or rows 	
Select column	<p>Click on a cell and select the Table Select Current Column menu option; the whole column is highlighted and selected for further formatting.</p>	
Show / hide table	<p>Click on a table cell and select the Table Show Gridlines menu</p>	

To	Action	See also
outline	option; this displays or hides the grid lines around the table cells. The grid lines are for display purpose only and do not appear on the printed document.	

18.1.2.2.13 Sections and Columns

The editor enables you to divide a document into multiple sections. A multiple section document is useful to:

- Vary the page margins from one page to another
- Create multiple columns of text

To	Action	See also
Create a new section	Select the Insert Insert Break Section Break menu option. This creates a new section on a new page. This option is not available when Edit Edit Page Header/ Footer is active.	
Edit the section parameters	Select the Edit Edit Section menu option; the Section Parameters dialog displays. Define: <ul style="list-style-type: none"> • The number of columns and column spacing - text in a multiple column section wraps at the end of the column; when the text reaches the end of the page, it resumes in the next column <p>When Page Mode is off, the page contains a single very thin and long column; when Page Mode is on, the correct column layout is shown</p> <ul style="list-style-type: none"> • The orientation - Portrait or Landscape • Whether to start the new section on the next page • The direction of text flow • Any special printing characteristics for the section. <p>You can also define any special page margins by selecting the File Page Layout menu option.</p>	File and Print Options ^[227]
Delete a section break	Move the cursor onto the section break line and press (Delete).	
Create a column break	Move the cursor to the appropriate point in the text and	

To	Action	See also
	<p>select the Insert Insert Break Column Break menu option.</p> <p>Normally in a multiple column section, the text flows from the end of one column to the top of the next column; a column break forces the text to the next column before the current column is completely filled.</p> <p>A column break is indicated by a line with a 'dot and dash' pattern; to delete the column break, simply position the cursor on the column break line and press (Delete).</p>	

18.1.2.2.14 Stylesheets and Table of Contents

The editor supports *Character* and *Paragraph-type* stylesheet style items:

- The Character stylesheet style constitutes a set of character formatting attributes and is applied to a character string
- The Paragraph stylesheet style constitutes both a set of character formatting attributes and a set of paragraph formatting attributes, and is applied to one or more paragraphs

You can also include special, structured text in the document, such as **page number**, **date and time** and **text input** fields.

To	Action	See also
Use double-byte characters	<p>Select the Edit Inline Ime menu option.</p> <p>This enables you to enter double-byte characters without using an external IME application.</p>	
Create and edit styles	<p>Select the Edit Edit Style menu option.</p> <p>The Edit Stylesheet dialog displays.</p> <p>Select the appropriate radio button to define a character style or a paragraph style.</p> <p>Either select an existing style to modify from the list box, or type in a name for a new style; click on the OK button to begin recording the style properties.</p> <p>You can use the ruler, toolbar or menu selections to modify the style items; these also reflect the currently-selected properties for the stylesheet item - please note that the paragraph properties are enabled only for the paragraph stylesheet items.</p> <p>After you have defined the required style, either select the Edit Edit Style menu option again or click anywhere in the document:</p> <ul style="list-style-type: none"> • If you modified an existing stylesheet item, the document automatically reflects the updated style • If you created a new stylesheet item, you can apply the style to 	

To	Action	See also
	highlighted text by selecting the Font Style or Paragraph Style menu options.	
Apply character styles	Select the Font Style menu option to apply a stylesheet style to a highlighted character string.	
Apply paragraph styles	Select the Paragraph Style menu option to apply a stylesheet style to a highlighted paragraph or range of paragraphs.	
Insert a table of contents	<p>Create and apply the required heading styles using the Edit Edit Style menu option, as above.</p> <p>Move the cursor to the point at which to insert the table of contents and select the Insert Table of Contents menu option.</p> <p>The table of contents is automatically updated whenever repagination occurs.</p>	Repagination ^[2289]
Insert date and time fields	<p>Move the cursor to the point at which to insert the current date and time, and select the Insert Date and Time menu option.</p> <p>The Insert Current Date/Time dialog displays, from which you select the required date and time format.</p> <p>The date and time are automatically updated whenever the page text is refreshed.</p>	
Insert your own data fields	<p>Not Supported In The Enterprise Architect RTF Generator</p> <p>Move the cursor to the point at which to insert the data field and select the Insert Data Field menu option.</p> <p>The Data Field Parameters dialog displays, in which you enter the field name and data value.</p>	
Insert your own text entry field	<p>Move the cursor to the point at which to insert the text entry field and select the Insert Text Input Field menu option.</p> <p>The Input Field Parameters dialog displays, in which you enter:</p> <ul style="list-style-type: none"> • The field name • The initial value to display as a default • The maximum length of the field • The text font in which to display the text <p>You also specify whether or not the field has a border.</p> <p>For example: Hel l o { i n s e r t n a m e f i e l d } .</p>	
Insert a	Move the cursor to the point at which to insert the checkbox and select	

To	Action	See also
selectable checkbox	<p>the Insert Checkbox Field menu option.</p> <p>The Checkbox Field Parameters dialog displays, in which you enter:</p> <ul style="list-style-type: none"> • The field name • Whether it defaults to selected • The size of the box surrounding the check 	
Define level numbering in generated document	<p>Select the Edit List and Overrides menu option.</p> <p>Set up the numbering list and the list overrides.</p> <p>Apply the numbering list to the headings set for packages and elements, using paragraph numbering.</p>	User defined list numbering [2299]

18.1.2.2.15 User-Defined Section Numbering

You might want to define the numbering format for the section levels in your generated RTF document.

For example:

- ```

1. Package Level 1
 1.1 Package Level 2 (child package)
 1.1.1 Element Level 1
 1.1.1.1 Element (child element)

```

To define the numbering format you:

- First create a numbering list and
- Then create a set of list overrides for this list

The overrides must also have the initial 1.0.0 setting altered to 1.1.1. You can then apply the numbering list to the headings set for packages and elements, using paragraph numbering.

(For further information on applying continuous section numbering throughout a **Virtual Document**, see *Section Numbering in Virtual Documents*.)

#### How to

To define the numbering format

| Step     | Action                                                                                                                                                 | See also |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>1</b> | <p>In the Template Editor, select the <b>Edit   List and Overrides   Create List Item</b> menu option.</p> <p>The List Properties dialog displays.</p> |          |
| <b>2</b> | <p>In the <b>List Name</b> field, type a name for the list.</p> <p>Click on the <b>OK</b> button to close the dialog.</p>                              |          |



| Step | Action                                                                                                                                                                                                                                                                                                                                                              | See also |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 3    | Select the <b>Edit   List and Overrides   Create List Override</b> menu option.<br>The List Override Properties dialog displays.                                                                                                                                                                                                                                    |          |
| 4    | In the <b>List to Override</b> field, type or select the name of the list you have just created.<br>Click on the <b>OK</b> button to close the dialog.                                                                                                                                                                                                              |          |
| 5    | To set the list level properties for each level, select the <b>Edit   List and Overrides   Edit List Level</b> menu option.<br>The List Level Properties dialog displays.                                                                                                                                                                                           |          |
| 6    | To set the first level numbering (used in the Package Section), select the <b>List override</b> radio button and type or select the list override item you have just created.                                                                                                                                                                                       |          |
| 7    | Ensure that the <b>List Level</b> field is set to <b>1</b> (for Packages) and the <b>Number text</b> field is set to <b>~1~</b> .<br>Click on the <b>OK</b> button to save the values and close the dialog.                                                                                                                                                         |          |
| 8    | Open the dialog again ( <b>Edit   List and Overrides   Edit List Level</b> ) and set: <ul style="list-style-type: none"> <li>• <b>List Level</b> to <b>2</b> (for the <i>Element Section</i> or <i>Child Package Section</i>, for example)</li> <li>• <b>Start at</b> to <b>1</b> (to ensure that numbering at this level begins at 1.1 rather than 1.0)</li> </ul> |          |
| 9    | Click on the <b>OK</b> button to close the dialog and save the changes.                                                                                                                                                                                                                                                                                             |          |
| 10   | Repeat steps 5 to 9 as required, incrementing the list level number and resetting <b>Start at</b> to <b>1</b> each time.                                                                                                                                                                                                                                            |          |

To apply the numbering levels you have defined (above), follow the steps below

| Step | Action                                                                                                                     | See also |
|------|----------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | In the Content window of the Template Editor, select the first item of text to be numbered (for example, <i>Package</i> ). |          |

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See also |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|      | <pre> package-&gt;  Package: {Pkg.Name} ¶ element-&gt;  Element: {Element.Name} ¶ child-elements-&gt;  &lt;child-elements ¶ &lt;element ¶ child-packages-&gt;  &lt;child-packages ¶ &lt;package ¶ </pre>                                                                                                                                                                                                                                                                                                                                                               |          |
| 2    | <p>Set the text style, using the style drop-down field in the Template Editor toolbar.</p> <pre> package-&gt;  <b>Package: {Pkg.Name} ¶</b> </pre>                                                                                                                                                                                                                                                                                                                                                                                                                     |          |
| 3    | <p>Click on the <b>Paragraph   List Numbering</b> menu option.<br/>The Apply Paragraph Numbering Using Lists dialog displays.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                      |          |
| 4    | <p>Select the required Numbering List and Override, and set the <b>Level</b> field to the required level (1, for the top level).<br/>Click on the <b>OK</b> button to close the dialog, and check that the required level has been applied to the selected text.</p> <pre> package-&gt;  <b>1. Package: {Pkg.Name} ¶</b> </pre>                                                                                                                                                                                                                                        |          |
| 5    | <ul style="list-style-type: none"> <li>Repeat steps 1 to 4 for the next level (Element), but at step 4 change the <b>Level</b> field to 2.</li> </ul> <pre> element-&gt;  <b>1.1 Element: {Element.Name} ¶</b> </pre>                                                                                                                                                                                                                                                                                                                                                  |          |
| 6    | <p>Continue applying the overrides for lower section levels as necessary, then save the template and generate your RTF report.<br/>The output should now resemble the following example:</p> <pre> 1.    <b>Package: Formal Requirements</b> 1.1  <b>Package: Manage Users</b> 1.1.1 <b>Element: REQ011 - Manage User Accounts</b> 1.1.2 <b>Element: REQ016 - Add Users</b> 1.2  <b>Package: Manage Inventory</b> 1.2.1 <b>Element: REQ019 - Manage Inventory</b> 1.2.2 <b>Element: REQ020 - Receive Books</b> 1.2.3 <b>Element: REQ021 - List Stock Levels</b> </pre> |          |

| Step | Action | See also |
|------|--------|----------|
|      |        |          |

### Learn more

- [Section Numbering in Virtual Documents](#)<sup>[2337]</sup>
- [Virtual Document](#)<sup>[2327]</sup>

#### 18.1.2.2.16 Frames and Drawing Objects

A frame is a rectangular area that can contain both text and pictures on the page. The text *outside* the frame flows around it. A drawing object can be a text box, rectangle or a line. The drawing object *overlays* the text around it. You can see frames and drawing objects only if **View | Page Mode** is selected. However, the *content* of a frame or text box is still visible if **Page Mode** is de-selected.

| To                                                                      | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | See also |
|-------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Embed a frame or drawing object at the cursor position</b>           | Select the <b>Insert   Frame</b> or <b>Insert   Drawing Object</b> menu option.                                                                                                                                                                                                                                                                                                                                                                                                               |          |
| <b>Insert text into the frame or drawing object text box</b>            | Click inside the outline and type the text at the cursor position.                                                                                                                                                                                                                                                                                                                                                                                                                            |          |
| <b>Rotate text to display it down the side of the frame or text box</b> | Select the <b>Edit   Edit Frame/Drawing Object   Rotate Text</b> menu option and select the text direction.                                                                                                                                                                                                                                                                                                                                                                                   |          |
| <b>Insert a picture into a frame or drawing object text box</b>         | Copy the picture, click inside the outline and paste the picture at the cursor position.                                                                                                                                                                                                                                                                                                                                                                                                      |          |
| <b>Size a frame or drawing object</b>                                   | Click inside the outline, click on a sizing tab on the outline and drag the tab to the required position.<br><br>If the frame or text box contains only a picture, the picture size is automatically adjusted to fill the outline; any text inside the outline is automatically wrapped to adjust to the new width.<br><br>In a frame the frame height is automatically adjusted, if necessary, to enclose all lines; in a text box, you must adjust the height manually to enclose the text. |          |
| <b>Move the frame or drawing object</b>                                 | Click inside the outline and then move the cursor just outside the outline so that the cursor changes to a plus-shape.<br><br>Drag the plus shape (and hence the outline) to the new location.                                                                                                                                                                                                                                                                                                |          |
| <b>Edit the</b>                                                         | Click on the outline and select the <b>Edit   Edit Frame/Drawing Object  </b>                                                                                                                                                                                                                                                                                                                                                                                                                 |          |

| To                                                                                                  | Action                                                                                                                                                                                                                                                                      | See also |
|-----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| relationship between a frame or drawing object and a point on the page (the vertical base position) | <p><b>Vertical Base Position</b> menu option.</p> <p>Select the point to lock the outline to; outlines locked to the top of the page or the top of the margin retain their vertical position when you insert text before them.</p>                                          |          |
| Edit the border and the background of a drawing object                                              | <p>Select the <b>Edit   Edit Frame/Drawing Object   Edit Drawing Object</b> menu option.</p> <p>On the Line and Fill Attributes dialog, select the options for the preferred border, line color, fill color and wrapping effect on the template text.</p>                   |          |
| Delete a frame or drawing object                                                                    | <p>Click on the outline and press ( <b>Delete</b> ).</p> <p>The editor prompts you to confirm the deletion; click on the <b>OK</b> button.</p> <p>Note that the deletion is actually reversible - press ( <b>Ctrl+Z</b> ) or select the <b>Edit   Undo</b> menu option.</p> |          |

#### 18.1.2.2.17 Search and Replace Commands

The first three menu options below all invoke the Search String Parameters dialog, if no search term has been defined.

Specify the term to search for, whether to search from the start of the file or forwards or backwards from the current cursor position, and whether the search should exactly match the case of the search term.

| To                                                                           | Action                                                                                                                                                                                                                                              | See also |
|------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Search for a text string                                                     | <p>Select the <b>Other   Search</b> menu option, or press ( <b>F5</b> ).</p> <p>The Search String Parameters dialog displays.</p> <p>The editor searches for the first instance of the specified character string as defined by the parameters.</p> |          |
| Find the next instance of a previously-defined text string in the file       | <p>Select the <b>Other   Search Forward</b> menu option, or press ( <b>Ctrl+F</b> ).</p> <p>The editor searches forwards for the next instance of the specified text string in the file, and highlights it.</p>                                     |          |
| Find the previous instance of the previously-defined text string in the file | <p>Select the <b>Other   Search Backward</b> menu option, or press ( <b>Ctrl+Shift+F</b> ).</p> <p>The editor searches backwards for the previous instance of the text string in the file, and highlights it.</p>                                   |          |
| Replace a text string                                                        | <p>Select the <b>Other   Replace</b> menu option, or press ( <b>F6</b> ).</p> <p>The Replace String Parameters dialog displays, in which you specify:</p>                                                                                           |          |

| To | Action                                                                                                                                                                                                                                                                             | See also |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|    | <ul style="list-style-type: none"> <li>• The text string to locate</li> <li>• The text string to replace it with</li> <li>• Whether to search the whole document or a highlighted block of text</li> <li>• Whether to confirm each replacement before making the change</li> </ul> |          |

### 18.1.2.3 Setting Sections for Reporting

Sections in RTF reports allow you to define what content to deliver in your report.



#### How to

To select model components to be documented in the report, using the RTF Style Template Editor

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See also                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Expand the Sections tree on the <template name> screen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 2    | <p>Select the checkbox next to the component name; the component name is then displayed as a section tag in the Content panel.</p> <p>More specific guidance is provided for selecting the following components:</p> <ul style="list-style-type: none"> <li>• Linked Documents and Document Artifact Contents</li> <li>• Tabular Sections</li> <li>• Embedded Sections</li> <li>• Child Sections</li> <li>• Constraint and Scenario Sections</li> <li>• Reporting elements held in external packages</li> <li>• Tagged Values</li> <li>• Profiled Relationship Matrices</li> </ul> | <p><a href="#">Reporting Linked Documents</a> <sup>[2309]</sup></p> <p><a href="#">Tabular Sections</a> <sup>[2305]</sup></p> <p><a href="#">Embedded Elements Sections</a> <sup>[2308]</sup></p> <p><a href="#">Child Sections</a> <sup>[2307]</sup></p> <p><a href="#">Constraint and Scenario Sections</a> <sup>[2310]</sup></p> <p><a href="#">Report Elements From External Packages</a> <sup>[2312]</sup></p> <p><a href="#">Tagged Value Sections</a> <sup>[2314]</sup></p> <p><a href="#">Reporting Profiled Relationship Matrices</a> <sup>[2311]</sup></p> |
| 3    | <p>In the Content Panel add text and fields to each report component.</p> <p>The position of the section tags within the Sections tree determines the position of the model component in the Content panel. For encapsulated components, selecting a child component automatically selects the parent also.</p>                                                                                                                                                                                                                                                                    | <p><a href="#">Add Section Content</a> <sup>[2315]</sup></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

### How to

To move a model component to a different position in the documentation template, follow the steps below.

| Step | Action                                                                                                                                                                                                                                 | See also |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | Select the component in the Sections panel.                                                                                                                                                                                            |          |
| 2    | Click on  and  to move the component up and down the Sections panel. |          |

### Notes

- To enable you to include elements from external packages, if the *Package.Diagram.Element* checkboxes are selected, you must select either the **Document each contained element in RTF** checkbox in the Diagram Properties dialog for at least one included diagram, or the **Include all diagram elements in report** checkbox on the Generate RTF Documentation dialog

### Learn more

- [Diagram Options](#) <sup>[2255]</sup>
- [Generate RTF Documentation](#) <sup>[2256]</sup>

### Learning Center topics

- (Alt+F1) | [Reporting | RTF Templates | Add a Section](#)

#### 18.1.2.3.1 Tabular Sections

The RTF Style Template Editor supports rendering a document section as a table. A tabular section is defined as a table containing any number of columns, but with only *two* rows:

- The first row is used to describe the headings of the columns, which you type in and format yourself
- The second row defines the output, which you specify by right-clicking in each cell and selecting the output type from the field list; the output is then generated iteratively for every occurrence of the section in question

### Topics

| Topic                          | Detail                                                                                                 | See also |
|--------------------------------|--------------------------------------------------------------------------------------------------------|----------|
| <b>Example Tabular Section</b> | In the following example, the <b>Model &gt; Glossary &gt;</b> section is defined as a tabular section: |          |

| Topic              | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | See also                                                                                                                                                                                         |      |         |                    |          |                                                                                                   |             |           |                                                                                                                                                                                                  |  |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------|--------------------|----------|---------------------------------------------------------------------------------------------------|-------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
|                    | <pre> model &gt; Model Glossary glossary &gt;   Term           Type           Meaning   {ModelGlossary.Term} {ModelGlossary.Type} {ModelGlossary.Meaning} &lt;/glossary&gt; &lt;/model&gt;                     </pre> <p>This renders the following document output:</p> <p><b>Model Glossary</b></p> <table border="1"> <thead> <tr> <th>Term</th> <th>Type</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>Accounting Periods</td> <td>Business</td> <td>A defined period of time whereby performance reports may be extracted. (normally 4 week periods).</td> </tr> <tr> <td>Association</td> <td>Technical</td> <td>A relationship between two or more entities. Implies a connection of some type - for example one entity uses the services of another, or one entity is connected to another over a network link.</td> </tr> </tbody> </table> | Term                                                                                                                                                                                             | Type | Meaning | Accounting Periods | Business | A defined period of time whereby performance reports may be extracted. (normally 4 week periods). | Association | Technical | A relationship between two or more entities. Implies a connection of some type - for example one entity uses the services of another, or one entity is connected to another over a network link. |  |
| Term               | Type                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Meaning                                                                                                                                                                                          |      |         |                    |          |                                                                                                   |             |           |                                                                                                                                                                                                  |  |
| Accounting Periods | Business                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | A defined period of time whereby performance reports may be extracted. (normally 4 week periods).                                                                                                |      |         |                    |          |                                                                                                   |             |           |                                                                                                                                                                                                  |  |
| Association        | Technical                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | A relationship between two or more entities. Implies a connection of some type - for example one entity uses the services of another, or one entity is connected to another over a network link. |      |         |                    |          |                                                                                                   |             |           |                                                                                                                                                                                                  |  |

**Notes**

- Under some circumstances, a table might repeat the header row rather than the output row; if this occurs, create another row in the table between the header row and the output row, and leave this blank
- If you type a carriage return between the end of the table and the section terminator, the table you generate has a line space between the rows

```

package">@
@
element">@
 * Element Name * Author *
 * {Element Name} * {Element Author} *
@
</element@
</package@

```

← Carriage Return

This generates the following table:

|  | Element Name     | Author       |
|--|------------------|--------------|
|  | Choose Recipient | John Redfern |

Line Space

|  |                 |                  |
|--|-----------------|------------------|
|  | Manage Contacts | Walter Frederick |
|--|-----------------|------------------|

To avoid this, ensure that there is no carriage return between the end of the table and the section terminator, as follows:

```

package">@
@
@
element">@
* Element.Name* Author*
* {Element.Name}* {Element.Author}*
<"element@
<"package@

```

← No Carriage Return

This generates a table with no space between the rows, as follows:

|  | Element Name     | Author           |
|--|------------------|------------------|
|  | Choose Recipient | John Redfern     |
|  | Manage Contacts  | Walter Frederick |

### 18.1.2.3.2 Child Sections

Child sections can be rendered in RTF documentation using one of the following two methods:

- Render model components directly into the RTF as defined by the section's content and fields
- Render indirectly to the RTF by using a parent section to describe the content.

The second option occurs as a result of creating a section that has a placeholder section tag (that is, no content within the tags). This method is used to create recursive documentation of child packages.

#### Topics

| Topic                                    | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also |
|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Example: Rendered Sub-section</b>     | <p>This example shows a template with content between the <i>Child Element</i> tags. In this example, child elements of the parent are rendered using the <i>Child Elements</i> section, because it contains valid content and fields.</p> <pre> element"&gt;{   {Element.Name}   Type: → → {Element.Type}--{Element.BaseClasses}   Status: → → {Element.Status}..Version: {Element.Version}..Phase: {Element.Phase}   Package: → → {Element.ParentPackage}--Keywords:..{Element.Tag}   Detail: → → Created on: {Element.DateCreatedShort}   {     {Element.Notes}   }   child*element"&gt;     → Child: {Element.Name}   &lt;child*element &lt;element </pre> |          |
| <b>Example: Non-rendered Sub-section</b> | <p>This example shows a template with no content between the child element tags; in this example, child elements of the parent are rendered using the element section, because the child element section is empty - the child element section is used as a placeholder.</p>                                                                                                                                                                                                                                                                                                                                                                                    |          |



| Topic                                                                  | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also                        |
|------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
|                                                                        | <pre> element&gt; * {Element.Name} Type: -&gt; -&gt; {Element.Type} -&gt; {Element.BaseClasses} Status: -&gt; -&gt; {Element.Status} -&gt; Version: {Element.Version} -&gt; Phase: {Element.Phase} Package: -&gt; -&gt; {Element.ParentPackage} -&gt; Keywords: -&gt; {Element.Tag} Detail: -&gt; -&gt; Created on: {Element.DateCreatedShort} {Element.Notes} child element&gt; * &lt;child element &lt;element                     </pre> |                                 |
| <b>Child Document Sections and Their Corresponding Parent Sections</b> | <b>Child Section</b>                                                                                                                                                                                                                                                                                                                                                                                                                        | <b>Section rendered when us</b> |
|                                                                        | Package->Child Package                                                                                                                                                                                                                                                                                                                                                                                                                      | Package                         |
|                                                                        | Package->Element->Child Element                                                                                                                                                                                                                                                                                                                                                                                                             | Package->Element                |
|                                                                        | Package->Element->Diagram                                                                                                                                                                                                                                                                                                                                                                                                                   | Package->Diagram                |
|                                                                        | Package->Diagram->Element                                                                                                                                                                                                                                                                                                                                                                                                                   | Package->Element                |
|                                                                        | Package->Diagram->Connector                                                                                                                                                                                                                                                                                                                                                                                                                 | Package->Element->Conne         |

**Notes**

- In principle, it is better to leave Child sections blank to replicate their parent sections

Child sections do not contain the same sub-section detail as their parents **Element::Child Element** does not contain sub-sections such as Scenario or Attribute, so where Child Element sections are populated, these sub-sections are not rendered.

An exception to this is cases where sub-sections are not required, but different formatting of the section fields is preferred.

**18.1.2.3.3 Embedded Elements Sections**

The RTF Report Template editor enables you to report on the following embedded elements on an element:

- ActivityParameter
- RequiredInterface
- ActionPin
- Port
- EntryPoint
- ExpansionNode
- ObjectNode
- ProvidedInterface

- ExitPoint
- Part

### Topics

| Topic                                    | Detail                                                                                                                                                                                                                                                            | See also |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Add Section For Embedded Elements</b> | <p>To report on the embedded elements on an element, select the <b>Package   Element   Embedded Elements</b> checkboxes.</p> <p>The embedded elements list is rendered into the RTF documentation at:</p> <pre>embedded elements &gt; &lt;embedded elements</pre> |          |

### Notes

- It is better to leave Embedded Elements sections blank to replicate their parent sections

Embedded Elements sections do not contain the same sub-section detail as their parents **Element::**

**Embedded Elements** does not contain sub-sections such as Scenario or Attribute, so where

Embedded Elements sections are populated, these sub-sections are not rendered.

### 18.1.2.3.4 Reporting Linked Documents

### Topics

| Topic                                                  | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See also                                                                                                                     |
|--------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| <b>Linked Documents and Document Artifact Contents</b> | <p>Linked documents and documents created via the Document Artifact element are rendered into RTF Documentation by selecting the <b>Linked Document</b> checkbox in the RTF Style Template Editor.</p> <p>The <b>Linked Document</b> checkbox is within the <b>Package   Element</b> and <b>Package   Package Element</b> hierarchies, towards the end; however, remember that checkboxes can be moved up and down the hierarchy to position information in the generated document as you require.</p> <p>In some templates, the <b>Linked Document</b> checkbox is only available as a child of the <b>External Requirements</b> checkbox.</p> <p>The linked document is rendered into the RTF documentation at:</p> <pre>linked document &gt; &lt;linked document</pre> | <p><a href="#">Linked Documents</a> <sup>[928]</sup></p> <p><a href="#">Creating Document Artifacts</a> <sup>[929]</sup></p> |

### 18.1.2.3.5 Reporting Profiled Relationship Matrices

The contents of a particular configuration of the Relationship Matrix can be saved as a profile, and these profiles can be included in RTF reports.

#### Topics

| Topic                               | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See also                                                                                                                                                |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Relationship Matrix Profiles</b> | All Relationship Matrix profiles in your project are listed in the Resources window, in the <i>Matrix Profiles</i> folder.                                                                                                                                                                                                                                                                                                                                                                                                                                       | <a href="#">Relationship Matrix</a> <sup>[639]</sup><br><a href="#">Matrix Profiles</a> <sup>[643]</sup><br><a href="#">Resources</a> <sup>[1003]</sup> |
| <b>Reporting Profiles</b>           | The report captures Relationship Matrix contents for the profile that includes the subject package as the source or target package. You do not directly specify a matrix profile; they are incorporated in the report by default.<br><br>If your report covers a tree of packages, only those profiles that include a source or target package in that tree are captured in the report.                                                                                                                                                                          |                                                                                                                                                         |
| <b>Render profiles</b>              | Relationship Matrix profiles are rendered into RTF Documentation by selecting the <b>Package   Relationship Matrix</b> checkbox in the RTF Style Template Editor; the matrix is rendered into the RTF documentation at:<br><br><code>&lt;relationship matrix &gt;</code><br><code>&lt;/relationship matrix</code><br><br>By right-clicking between the markers, you can insert a number of fields to identify specific aspects of the profile, such as relationship name, direction, type, target element and element type, and source element and element type. |                                                                                                                                                         |

### 18.1.2.3.6 Constraint and Scenario Sections

Elements can be re-used in different diagrams across a model; this can often mean that a diagram contains elements that are held in packages other than the diagram's parent package.

When generating an RTF report on a package, you might want to include elements that are not held in that package, but that are referenced in diagrams within the package. To do this you must:

- Select the appropriate diagram options on the Generate RTF Documentation dialog and/or the Diagram page of the diagram Properties dialog
- Enable the appropriate template sections in the report template, as described in this topic

**Access** [Project Browser package context menu | Documentation | Rich Text Format \(RTF\) Report \(F8\)](#)

#### How to

To enable the template sections

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See also |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | Select the <b>Rich Text Format (RTF) Report</b> menu option.<br>The Generate RTF Documentation dialog displays.                                                                                                                                                                                                                                                                                                                                        |          |
| 2    | In the <b>Use Template</b> field, click on the drop-down arrow and select the appropriate customized template (not a system-provided one).                                                                                                                                                                                                                                                                                                             |          |
| 3    | Click on the <b>Edit Template</b> button to display the RTF Template Editor.                                                                                                                                                                                                                                                                                                                                                                           |          |
| 4    | In the Sections panel on the left-hand side of the editor window, select the <b>Package::Element</b> checkbox.<br>Selecting the checkbox adds the following set of sections to the Content panel of the template: <pre data-bbox="336 931 778 1144"> package &gt; [right-click-to-insert-Package-field(s)] element &gt; [right-click-to-insert-Element-field(s)] child elements &gt; &lt; child elements &lt; element &lt; package           </pre>    |          |
| 5    | Delete the <i>[right-click-to-insert-Element-field(s)]</i> text to leave a blank area.<br>Right-click and select the <b>Insert Field   Name</b> context menu option, this inserts an <b>Element.Name</b> field in the template.                                                                                                                                                                                                                        |          |
| 6    | In the Sections panel, select the <b>Package::Diagram::Element</b> checkbox.<br>Selecting the checkbox adds the following set of sections to the Content panel of the template: <pre data-bbox="336 1525 799 1771"> package-&gt;¶ [right-click-to-insert-Package-field(s)]¶ diagram-&gt;¶ [right-click-to-insert-Diagram-field(s)]¶ element-&gt;¶ [right-click-to-insert-Element-field(s)]¶ &lt;-element¶ &lt;-diagram¶ &lt;-package¶           </pre> |          |
| 7    | To report on the linked elements in: <ul data-bbox="373 1890 1206 1962" style="list-style-type: none"> <li>the same style as defined in the <b>Package::Elements</b> section, delete the <i>[right-click-to-insert-Element-field(s)]</i> text to leave a blank area</li> </ul> NB: <b>Package::Diagram::Element</b> , if left blank, replicates the format of                                                                                          |          |

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See also |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|      | <p><b>Package::Element</b> including sub-element sections (for example, <b>Package::Element::Scenario</b>), but excluding the <b>Child Element</b> section.</p> <ul style="list-style-type: none"> <li>a different style to the elements within the selected package, set up the style as appropriate</li> </ul> <p>If using the Legacy RTF generator, the <i>Element</i> checkbox is automatically checked in your customized template, when you select the <b>Document each contained element in RTF</b> checkbox.</p> <p><b>Package::Diagram::Element</b> does not have an option to add these sections.</p> |          |

#### Learn more

- [Diagram Options](#) 

#### Learning Center topics

- (Alt+F1) | Reporting | RTF Templates | Reporting Elements from External Packages**

#### **18.1.2.3.7 Report Elements From External Packages**

Elements can be re-used in different diagrams across a model; this can often mean that a diagram contains elements that are held in packages other than the diagram's parent package.

When generating an RTF report on a package, you might want to include elements that are not held in that package, but that are referenced in diagrams within the package. To do this you must:

- Select the appropriate diagram options on the Generate RTF Documentation dialog and/or the Diagram page of the diagram Properties dialog
- Enable the appropriate template sections in the report template, as described in this topic

**Access Project Browser package context menu | Documentation | Rich Text Format (RTF) Report (F8)**

#### How to

To enable the template sections

| Step | Action                                                                                                                                     | See also |
|------|--------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | Select the <b>Rich Text Format (RTF) Report</b> menu option.<br>The Generate RTF Documentation dialog displays.                            |          |
| 2    | In the <b>Use Template</b> field, click on the drop-down arrow and select the appropriate customized template (not a system-provided one). |          |

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See also |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 3    | Click on the <b>Edit Template</b> button to display the RTF Template Editor.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |          |
| 4    | <p>In the Sections panel on the left-hand side of the editor window, select the <b>Package::Element</b> checkbox.</p> <p>Selecting the checkbox adds the following set of sections to the Content panel of the template:</p> <pre data-bbox="339 600 778 813"> package &gt; [right-click-to-insert-Package-field(s)] element &gt; [right-click-to-insert-Element-field(s)] child elements &gt; &lt; child elements &lt; element &lt; package </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |          |
| 5    | <p>Delete the <i>[right-click-to-insert-Element-field(s)]</i> text to leave a blank area.</p> <p>Right-click and select the <b>Insert Field   Name</b> context menu option, this inserts an <b>Element.Name</b> field in the template.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |          |
| 6    | <p>In the Sections panel, select the <b>Package::Diagram::Element</b> checkbox.</p> <p>Selecting the checkbox adds the following set of sections to the Content panel of the template:</p> <pre data-bbox="339 1149 802 1395"> package-&gt;¶ [right-click-to-insert-Package-field(s)]¶ diagram-&gt;¶ [right-click-to-insert-Diagram-field(s)]¶ element-&gt;¶ [right-click-to-insert-Element-field(s)]¶ &lt;-element¶ &lt;-diagram¶ &lt;-package¶ </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |          |
| 7    | <p>To report on the linked elements in:</p> <ul data-bbox="371 1507 1209 1776" style="list-style-type: none"> <li>the same style as defined in the <b>Package::Elements</b> section, delete the <i>[right-click-to-insert-Element-field(s)]</i> text to leave a blank area</li> </ul> <p>NB: <b>Package::Diagram::Element</b>, if left blank, replicates the format of <b>Package::Element</b> including sub-element sections (for example, <b>Package::Element::Scenario</b>), but excluding the <b>Child Element</b> section.</p> <ul data-bbox="371 1709 1241 1776" style="list-style-type: none"> <li>a different style to the elements within the selected package, set up the style as appropriate</li> </ul> <p>If using the Legacy RTF generator, the <i>Element</i> checkbox is automatically checked in your customized template, when you select the <b>Document each contained element in RTF</b> checkbox.</p> <p><b>Package::Diagram::Element</b> does not have an option to add these sections.</p> |          |

Learn more

- [Diagram Options](#)<sup>[2255]</sup>

Learning Center topics

- (Alt+F1) | [Reporting | RTF Templates](#) | [RTF Templates](#) | [Reporting Elements from External Packages](#)

**18.1.2.3.8 Tagged Value Sections**

Tagged Values can be reported:

- As a group using a section, and
- Individually using **valueOf()** fields

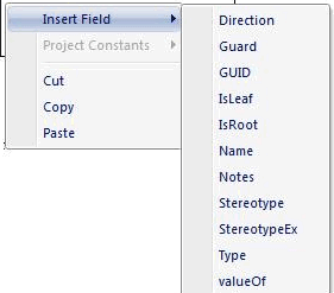
The following objects and connectors support Tagged Values:

- Package Element
- Element
- Connector (and Source and Target elements)
- Attribute
- Operation (Method)
- External Requirement

Each of these has an associated section in the RTF template editor, which includes both a Tagged Value subsection and the ability to insert a **valueOf()** field.

Topics

| Topic                       | Detail                                                                                                                                                                                                                                                                                 | See also                                              |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| <b>Tagged Value Section</b> | The Tagged Value section reports all Tagged Values for the object or connector.<br><br>tagged value ><br>[right-click-to-insert-Tagged-Value-field(s)]<br>< tagged value                                                                                                               |                                                       |
| <b>The valueOf Field</b>    | For the objects defined above you can add a <b>valueOf()</b> field to report a specific Tagged Value.<br><br>To insert the field, right-click between the appropriate set of section tags and select the <b>Insert Field   valueOf</b> option, as shown for a Connector section below: | <a href="#">Add Section Content</a> <sup>[2315]</sup> |

| Topic                              | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See also                           |                                      |        |       |                               |                          |                          |                                |                               |                                 |                                 |                              |                                    |                                |                                |                                    |  |                                    |                                    |                                      |  |                |                           |                              |  |                           |                             |  |  |                             |                           |  |  |                           |                          |  |  |                          |  |  |  |                             |  |  |  |
|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|--------------------------------------|--------|-------|-------------------------------|--------------------------|--------------------------|--------------------------------|-------------------------------|---------------------------------|---------------------------------|------------------------------|------------------------------------|--------------------------------|--------------------------------|------------------------------------|--|------------------------------------|------------------------------------|--------------------------------------|--|----------------|---------------------------|------------------------------|--|---------------------------|-----------------------------|--|--|-----------------------------|---------------------------|--|--|---------------------------|--------------------------|--|--|--------------------------|--|--|--|-----------------------------|--|--|--|
|                                    | <p><code>connector &gt;</code></p> <p><i>Connections</i></p> <table border="1"> <thead> <tr> <th>Connector</th> <th>Source</th> <th>Target</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td><code>{Connector.Type}</code></td> <td><code>source &gt;</code></td> <td><code>target &gt;</code></td> <td><code>{Connector.Notes}</code></td> </tr> <tr> <td><code>{Connector.Name}</code></td> <td><code>{ConnSource.Scope}</code></td> <td><code>{ConnTarget.Scope}</code></td> <td><code>constraint &gt;</code></td> </tr> <tr> <td><code>{Connector.Direction}</code></td> <td><code>{ConnSource.Role}</code></td> <td><code>{ConnTarget.Role}</code></td> <td><code>{ConnConstraint.Type}</code></td> </tr> <tr> <td></td> <td><code>{ConnSource.RoleNote}</code></td> <td><code>{ConnTarget.RoleNote}</code></td> <td><code>{{ConnConstraint.Name}}</code></td> </tr> <tr> <td></td> <td><code>}</code></td> <td><code>element &gt;</code></td> <td><code>&lt; constraint</code></td> </tr> <tr> <td></td> <td><code>element &gt;</code></td> <td><code>{Element.Name}</code></td> <td></td> </tr> <tr> <td></td> <td><code>{Element.Name}</code></td> <td><code>&lt; element</code></td> <td></td> </tr> <tr> <td></td> <td><code>&lt; element</code></td> <td><code>&lt; target</code></td> <td></td> </tr> <tr> <td></td> <td><code>&lt; source</code></td> <td></td> <td></td> </tr> <tr> <td></td> <td><code>&lt; connector</code></td> <td></td> <td></td> </tr> </tbody> </table>  <p>When you select the <b>valueOf</b> field from the context menu, the template editor prompts you to specify the Tagged Value from which to extract the value for the report output; this tag should be one of the tags associated with the model component, such as <i>ConnectorAltName</i> for a connector.</p> <p>When you provide the tag name, the template editor adds the field at the cursor position, in the format:</p> <pre>{Connector.valueOf(tagname)}</pre> <p>For example:</p> <pre>{Connector.valueOf(ConnectorAltName)}</pre> <p>For clarity, you could type some lead-in text or the meaning of the Tagged Value immediately preceding the value field</p> <p><b>Alternative Name:</b> <code>{Connector.valueOf(ConnectorAltName)}</code></p> | Connector                          | Source                               | Target | Notes | <code>{Connector.Type}</code> | <code>source &gt;</code> | <code>target &gt;</code> | <code>{Connector.Notes}</code> | <code>{Connector.Name}</code> | <code>{ConnSource.Scope}</code> | <code>{ConnTarget.Scope}</code> | <code>constraint &gt;</code> | <code>{Connector.Direction}</code> | <code>{ConnSource.Role}</code> | <code>{ConnTarget.Role}</code> | <code>{ConnConstraint.Type}</code> |  | <code>{ConnSource.RoleNote}</code> | <code>{ConnTarget.RoleNote}</code> | <code>{{ConnConstraint.Name}}</code> |  | <code>}</code> | <code>element &gt;</code> | <code>&lt; constraint</code> |  | <code>element &gt;</code> | <code>{Element.Name}</code> |  |  | <code>{Element.Name}</code> | <code>&lt; element</code> |  |  | <code>&lt; element</code> | <code>&lt; target</code> |  |  | <code>&lt; source</code> |  |  |  | <code>&lt; connector</code> |  |  |  |
| Connector                          | Source                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Target                             | Notes                                |        |       |                               |                          |                          |                                |                               |                                 |                                 |                              |                                    |                                |                                |                                    |  |                                    |                                    |                                      |  |                |                           |                              |  |                           |                             |  |  |                             |                           |  |  |                           |                          |  |  |                          |  |  |  |                             |  |  |  |
| <code>{Connector.Type}</code>      | <code>source &gt;</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <code>target &gt;</code>           | <code>{Connector.Notes}</code>       |        |       |                               |                          |                          |                                |                               |                                 |                                 |                              |                                    |                                |                                |                                    |  |                                    |                                    |                                      |  |                |                           |                              |  |                           |                             |  |  |                             |                           |  |  |                           |                          |  |  |                          |  |  |  |                             |  |  |  |
| <code>{Connector.Name}</code>      | <code>{ConnSource.Scope}</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | <code>{ConnTarget.Scope}</code>    | <code>constraint &gt;</code>         |        |       |                               |                          |                          |                                |                               |                                 |                                 |                              |                                    |                                |                                |                                    |  |                                    |                                    |                                      |  |                |                           |                              |  |                           |                             |  |  |                             |                           |  |  |                           |                          |  |  |                          |  |  |  |                             |  |  |  |
| <code>{Connector.Direction}</code> | <code>{ConnSource.Role}</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <code>{ConnTarget.Role}</code>     | <code>{ConnConstraint.Type}</code>   |        |       |                               |                          |                          |                                |                               |                                 |                                 |                              |                                    |                                |                                |                                    |  |                                    |                                    |                                      |  |                |                           |                              |  |                           |                             |  |  |                             |                           |  |  |                           |                          |  |  |                          |  |  |  |                             |  |  |  |
|                                    | <code>{ConnSource.RoleNote}</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <code>{ConnTarget.RoleNote}</code> | <code>{{ConnConstraint.Name}}</code> |        |       |                               |                          |                          |                                |                               |                                 |                                 |                              |                                    |                                |                                |                                    |  |                                    |                                    |                                      |  |                |                           |                              |  |                           |                             |  |  |                             |                           |  |  |                           |                          |  |  |                          |  |  |  |                             |  |  |  |
|                                    | <code>}</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <code>element &gt;</code>          | <code>&lt; constraint</code>         |        |       |                               |                          |                          |                                |                               |                                 |                                 |                              |                                    |                                |                                |                                    |  |                                    |                                    |                                      |  |                |                           |                              |  |                           |                             |  |  |                             |                           |  |  |                           |                          |  |  |                          |  |  |  |                             |  |  |  |
|                                    | <code>element &gt;</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <code>{Element.Name}</code>        |                                      |        |       |                               |                          |                          |                                |                               |                                 |                                 |                              |                                    |                                |                                |                                    |  |                                    |                                    |                                      |  |                |                           |                              |  |                           |                             |  |  |                             |                           |  |  |                           |                          |  |  |                          |  |  |  |                             |  |  |  |
|                                    | <code>{Element.Name}</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | <code>&lt; element</code>          |                                      |        |       |                               |                          |                          |                                |                               |                                 |                                 |                              |                                    |                                |                                |                                    |  |                                    |                                    |                                      |  |                |                           |                              |  |                           |                             |  |  |                             |                           |  |  |                           |                          |  |  |                          |  |  |  |                             |  |  |  |
|                                    | <code>&lt; element</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <code>&lt; target</code>           |                                      |        |       |                               |                          |                          |                                |                               |                                 |                                 |                              |                                    |                                |                                |                                    |  |                                    |                                    |                                      |  |                |                           |                              |  |                           |                             |  |  |                             |                           |  |  |                           |                          |  |  |                          |  |  |  |                             |  |  |  |
|                                    | <code>&lt; source</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                    |                                      |        |       |                               |                          |                          |                                |                               |                                 |                                 |                              |                                    |                                |                                |                                    |  |                                    |                                    |                                      |  |                |                           |                              |  |                           |                             |  |  |                             |                           |  |  |                           |                          |  |  |                          |  |  |  |                             |  |  |  |
|                                    | <code>&lt; connector</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                    |                                      |        |       |                               |                          |                          |                                |                               |                                 |                                 |                              |                                    |                                |                                |                                    |  |                                    |                                    |                                      |  |                |                           |                              |  |                           |                             |  |  |                             |                           |  |  |                           |                          |  |  |                          |  |  |  |                             |  |  |  |


### 18.1.2.3.9 Add Section Content

The RTF Style Template Editor uses pairs of tags to define the layout of the documentation content. To insert a pair of model component tags into the Content section of the Editor, use the Sections panel of the RTF Style Template Editor.

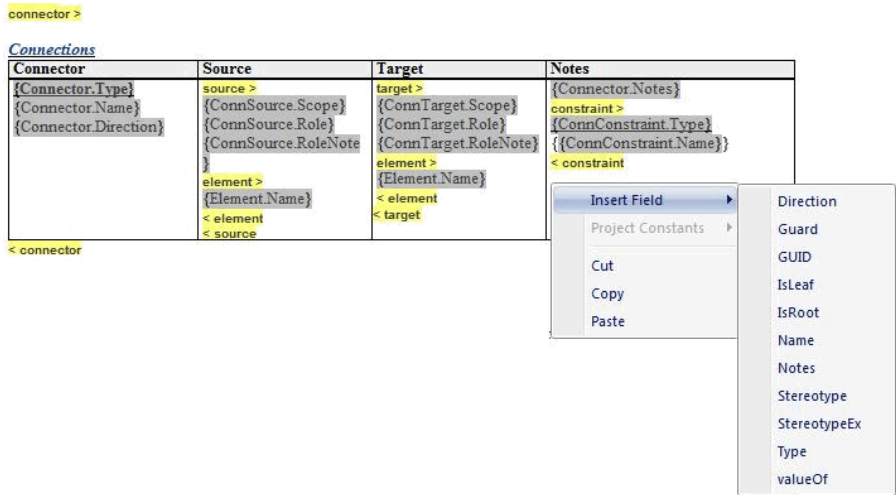
The beginning of a model component is represented by a yellow highlighted `sectionname >` tag; the end of the model component is represented by `<sectionname`.

#### How to

To add model component content, follow the steps below.

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                  | See also                                                                                                                       |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| 1    | <p>Right-click in the area between the opening and closing tags; this displays a context menu that enables you to:</p> <ul style="list-style-type: none"> <li>perform simple text editing (cut, copy and paste)</li> <li>select from a list of defined project constants to insert at the cursor position; if no project constants have been defined, the menu option is not</li> </ul> | <p><a href="#">Project Constants</a> </p> |



| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See also |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|      | <p>available</p> <ul style="list-style-type: none"> <li>select from a context-sensitive list of fields to add to this component section of the RTF Documentation; any additional information entered between the tags is also included in the generated RTF Documentation</li> </ul> <p>For example, in the <code>connector&gt;</code> section shown below, the user has right-clicked in the fourth column of the table, underneath the <code>&lt;constraint</code> tag; they can then select to add another field to the Notes on each connector listed in the table.</p>  |          |

**Notes**

- If you select a field with short date format (such as `Pkg.DateCreatedShort`, `Diagram.DateModifiedShort` or `Element.DateCreatedShort`) the format is actually drawn from the MS Windows settings

To use a different short date format, click on the **Start** icon on the Windows desktop and select the **Control Panel | Regional and Language Options | Customize** option

**18.1.2.4 Import RTF Template**

Enterprise Architect provides a number of RTF document templates, and enables you to create others. However, you might already have corporate formats and templates in use in your organization, so Enterprise Architect also enables you to import existing templates into the RTF Generator.

**How to**

To import an external template

| Step | Action                                         | See also |
|------|------------------------------------------------|----------|
| 1    | Save the external template as an RTF document. |          |

| Step | Action                                                                                                                                                                                                    | See also                                                            |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| 2    | In Enterprise Architect, create a new blank template; name the template but do not specify an existing template to copy from.                                                                             | <a href="#">RTF Templates</a> <sup>[2270]</sup>                     |
| 3    | When the template is listed on the Templates tab of the Generate RTF Documentation dialog, click on the name and click on the <b>Edit</b> button.<br>The RTF Document Template Editor dialog displays.    |                                                                     |
| 4    | Right-click on the document and select the <b>File   Import</b> context menu option.<br>The Microsoft Word File Open dialog displays.                                                                     |                                                                     |
| 5    | Locate your template RTF file, and click on the <b>Open</b> button.<br>The Open dialog closes, returning you to the Document Template Editor dialog; this now contains your imported template.            |                                                                     |
| 6    | Right-click and select the <b>File   Save</b> menu option.<br>If necessary, make any changes to the template and select <b>File   Save</b> again before selecting <b>File   Close</b> to exit the dialog. | <a href="#">RTF Style Template Editor Options</a> <sup>[2275]</sup> |

### Notes

- Standard graphical images (such as a logo in the header, main text or footer) are imported; however, any Word-based meta-file graphics are not imported
- You can select the new template to use in generating an RTF document, either on the Generate RTF Documentation dialog or in a Master Document or Model Document element

### Learn more

- [Generate RTF Documentation](#) <sup>[2253]</sup>
- [Master Document](#) <sup>[2330]</sup>
- [Model Document](#) <sup>[2331]</sup>

### Learning Center topics

- **(Alt+F1) | Reporting | RTF Template | Importing a Company Template**

### 18.1.3 The Legacy RTF Report Generator

#### Topics

| Topic           | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See also |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Abstract</b> | <p>Creating a <b>Rich Text Format</b> (RTF) document is a simple and flexible process. An RTF document is based on a package or an element in your project (more usually a package). To produce a document, you select the package or element to report on in the <b>Project Browser</b>, <b>Package Browser</b>, <b>Diagram List</b> or <b>Model Search</b>, then press ( <b>F8</b> ) to display the <b>Generate RTF Documentation</b> dialog. On the <b>Advanced</b> tab, click on the <b>Switch generator</b> button to access the Legacy <b>Rich Text Format Report</b> dialog.</p> <p>The <b>Rich Text Format Report</b> dialog enables you to set the exact contents and look and feel of your report. You enter the file name of the report, a heading, additional notes, template name (for saving the set-up) and other options. You can also select the style of the report; either plain or formal.</p> <p>Optionally, you can set up a filter, the details to include, element types to exclude, whether to process child packages, whether to show diagrams and the diagram format.</p> <p>You can switch back to the <b>Generate RTF Documentation</b> dialog by clicking on the <b>Switch RTF Generator</b> button.</p> |          |

#### Notes

- The Legacy Generator is available if you have RTF templates created in releases of Enterprise Architect prior to 4.1, and you prefer to generate RTF reports using the original generator. However, as you can generate reports from these templates using the post-Enterprise Architect 4.1 RTF Generator, the Legacy Generator and instructions for its use are no longer updated. However, reports produced using the Legacy RTF Generator do reflect the Notes formatting feature in any text associated with elements
- The Rich Text Format Report dialog panels are individually described in the subsequent topics of this section (listed below). The dialog has a lot of options; get to know them all to produce output at the level of detail suited to your project

#### Learn more

- [Document a Single Element](#)<sup>[2319]</sup>
- [Set the Main RTF Properties](#)<sup>[2319]</sup>
- [Apply a Filter](#)<sup>[2320]</sup>
- [Exclude Elements](#)<sup>[2321]</sup>
- [RTF Diagram Format](#)<sup>[2321]</sup>
- [Model Include](#)<sup>[2321]</sup>
- [RTF Report Options](#)<sup>[2322]</sup>
- [RTF Report Selections](#)<sup>[2323]</sup>
- [Generate the Report](#)<sup>[2323]</sup>
- [Diagrams Only Report](#)<sup>[2340]</sup>
- [Report Templates](#)<sup>[2323]</sup>
- [Include or Exclude a Package from Report](#)<sup>[2260]</sup>
- [Save as Document](#)<sup>[2325]</sup>
- [Generate RTF Documentation Dialog](#)<sup>[2256]</sup>
- [Notes Toolbar](#)<sup>[979]</sup>

### 18.1.3.1 Document a Single Element

RTF documentation can also be generated for a single element.

**Access** Select an element and then go to: **Element | Advanced | Rich Text Format (RTF) Report**

Click on the **Switch Generator** button to display the **Rich Text Format Report** dialog. See **The Legacy RTF Report Generator** and its related topics for further information.

#### Learn more

- [Generate RTF Documentation Dialog](#)<sup>[2256]</sup>
- [The Legacy RTF Report Generator](#)<sup>[2318]</sup>

### 18.1.3.2 Set the Main RTF Properties

The main section of the **Rich Text Format Report** dialog enables you to set the output location and appearance of the final RTF document.

#### How to

| Step | Action                                                                                                                | See also                                                          |
|------|-----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| 1.   | Open the <b>Rich Text Format Report</b> dialog (see <b>The Legacy RTF Report Generator</b> topic for how to do this). | <a href="#">The Legacy RTF Report Generator</a> <sup>[2318]</sup> |
| 2.   | Supply an <b>Output Filename</b> to save the report into; always include the extension .RTF as part of the filename.  |                                                                   |
| 3.   | Provide a <b>Template Name</b> to save this report set-up.                                                            |                                                                   |
| 4.   | Select a report <b>Style</b> : Formal or Basic.                                                                       |                                                                   |
| 5.   | Type a <b>Heading</b> for your report; this appears as the first heading item in your output.                         |                                                                   |
| 6.   | Select your required <b>Heading Style and Initial Heading Level Indent</b> from the drop-down lists.                  |                                                                   |

#### Notes

- It is recommended that you enter a full path name for your report. The images in your report are saved externally in an images directory, and supplying the full directory path avoids confusion over the

location of these images. Also, if you move your report you must also move the images directory

### 18.1.3.3 Apply a Filter

#### Topics

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See also |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Usage | <p>You can apply a filter on the <b>Rich Text Format Report</b> dialog to include or exclude elements by date modified, phase or status. This helps to track changes and break a document into multiple delivery phases.</p> <p>Open the <b>Rich Text Format Report</b> dialog (see <b>The Legacy RTF Report Generator</b> for how to do this).</p> <ul style="list-style-type: none"> <li>• To enable the date filter, select the checkbox in the date field.</li> <li>• In the first two <b>Only include objects</b> fields, click on the drop-down arrows and select the appropriate criteria (<b>Modified/ Created, Before/After</b>).</li> <li>• The package phase filter applies at the package level (not the element level) and ignores the phase of the root package that you are documenting. To enable the phase filter, in the <b>Where Package Phase</b> field click on the drop-down arrow and select an operator; Enterprise Architect filters out all packages that do not meet the selection criteria. All elements in the package are ignored, regardless of their individual phase.</li> <li>• The element status filter enables you to limit the output by element status. Unlike the package phase filter, this filter applies to every element. You can filter against a status of like or <i>unlike</i> a criterion, for example, <i>like proposed</i>, or against the <i>in</i> and <i>not in</i> operators, such as <i>in approved</i>, <i>not in validated</i>. When using the <i>in</i> and <i>not in</i> operators, enter a comma-separated list of status types as your criteria expression.</li> </ul> |          |

#### Learn more

- [The Legacy RTF Report Generator](#) 

### 18.1.3.4 Exclude Elements

#### Topics

| Topic        | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See also                                                          |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| <b>Usage</b> | <p>The Rich Text Format Report dialog enables you to exclude elements of any type from your final output; this is useful when you want to highlight particular items and not clutter up a report with too much detail.</p> <p>Open the Rich Text Format Report dialog (see <i>The Legacy RTF Report Generator</i> for details on how to do this).</p> <p>Click on each element to exclude, or click on the <b>All</b> button to exclude all elements.</p> <p>Click on the <b>None</b> button to clear your selections.</p> | <a href="#">The Legacy RTF Report Generator</a> <sup>[2318]</sup> |

### 18.1.3.5 RTF Diagram Format

#### Topics

| Topic        | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also                                                          |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| <b>Usage</b> | <p>You can output diagrams to Bitmap files, GIF files or Windows Metafiles.</p> <p>Open the Rich Text Format Report dialog (see <i>The Legacy RTF Report Generator</i> for details on how to do this).</p> <p>In the Diagram format panel (bottom center of the dialog) select the required format for the report.</p> <p>Generally the two metafile options (EMF and WMF) are recommended; however, there are times that the others might be suitable.</p> | <a href="#">The Legacy RTF Report Generator</a> <sup>[2318]</sup> |

### 18.1.3.6 Model Include

#### Topics

| Topic        | Detail                                                                                                                                                                                                                                                                                                                                                                             | See also                                                                                                                                               |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Usage</b> | <p>The Model Include panel of the Rich Text Format Report dialog has the following options:</p> <ul style="list-style-type: none"> <li>• Glossary to include the project glossary</li> <li>• Tasks to include project tasks</li> <li>• Issues to include project issues</li> </ul> <p>Select the appropriate checkbox to include the items in the generated RTF documentation.</p> | <a href="#">Project Glossary</a> <sup>[467]</sup><br><a href="#">Project Tasks</a> <sup>[460]</sup><br><a href="#">Project Issues</a> <sup>[462]</sup> |

### 18.1.3.7 RTF Report Options

Additional RTF report options you can select from the **Options** panel on the **Rich Text Format Report** dialog are shown below.

#### Topics

| Topic        | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also                                                                 |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| <b>Usage</b> | <p>You can select whether or not to recursively document packages, show diagrams or add a page break before each new package. Select the:</p> <ul style="list-style-type: none"> <li>• <b>Process all Children</b> checkbox to recursively process all child packages within the main package</li> <li>• <b>Show Diagrams</b> checkbox to include diagrams in your document. Clear this item for no diagrams</li> <li>• <b>New page per package</b> checkbox to force a page break on each new package (excepting empty packages)</li> <li>• <b>Document all elements</b> checkbox to include all elements included in the project</li> <li>• <b>Document Packages</b> checkbox to document the package as an element in addition to the documentation that would normally be produced for package documentation</li> <li>• <b>Hide 'note-less' elements</b> checkbox to exclude all elements without notes from the documentation</li> <li>• <b>Embed Diagrams in Document</b> checkbox to ensure that the diagram images are contained within the RTF document rather than stored in a linked external file</li> <li>• <b>Skip root package</b> checkbox to exclude the parent package from the documentation and include only the child packages</li> <li>• <b>Document Linked Elements</b> checkbox to include the object details for linked elements that do not originate from the selected package</li> <li>• <b>Use Heading styles for Details</b> checkbox to ensure that the details are formatted as heading styles rather than formatted text; this option is only available when the Heading Style field in the <b>Main section</b> of the <b>Rich Text Format Report</b> dialog is set to Max 9 levels - elements are package + 1.</li> </ul> | <a href="#">Setting the Main RTF Properties</a><br><small>[2319]</small> |

### 18.1.3.8 RTF Report Selections

The **For each Object Include** section of the **Rich Text Format Report** dialog enables you to select the documentation sections to include in your report.

#### Topics

| Topic        | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See also |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Usage</b> | <p>What you include or exclude governs how simple or detailed your report is. You can create multiple reports at different levels of detail for different audiences. Experiment with these options to see what effect inclusion or exclusion has. Most items are self-explanatory.</p> <p>Selecting the checkbox against a category item in the list selects all of the options that are contained in the category. To expand a category, click on the +symbol next to the category name. To exercise greater control over a category of options expand the top level item and then select the required individual items from the list.</p> <p>Sometimes an item applies only to a certain type of element; for example, <b>Attributes</b> only applies to Class elements and a few other element types. The <b>Child Diagrams</b> option shows or hides any diagrams that are attached under a model element; for example, a Use Case might have a Scenario diagram attached.</p> |          |

#### Notes

- Use this feature to produce the right level of detail for your audience. Technical readers might want to see everything, whilst management might require only the general outline

### 18.1.3.9 Generate the Report

#### Topics

| Topic        | Detail                                                                                                                                                                                                                                   | See also |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Usage</b> | <p>Once you have set up the document properties as required, click on the <b>Generate</b> button to generate the report.</p> <p>When you have generated the document, click on the <b>View</b> button to open the report in MS Word.</p> |          |

### 18.1.3.10 Legacy RTF Style Templates

The **Legacy RTF Style Editor** enables you to edit the RTF associated with various sections of the RTF Report facility in Enterprise Architect. You would typically use this functionality to customize a report's look and feel for your company or client.

If you have previously defined and saved a template, click on the **Load** button on the **Rich Text Format Report dialog** to open the list of defined templates. Select one in order to load it as the current template; all the features saved become the current features. This enables you to define a set of standard report types



that streamline document production.


**Access** [View](#) | [More Project Tools](#) | [Project Resources](#) | [Resources](#)

To create or Edit RTF Style Templates, follow the steps below

#### **How to**

| Step | Action                                                                                                                                                                                                                                              | See also |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | Select the <b>View   More Project Tools   Project Resources</b> menu option to display the <b>Resources</b> window.                                                                                                                                 |          |
| 2.   | Expand the <b>Templates</b> folder.                                                                                                                                                                                                                 |          |
| 3.   | To edit an <i>existing</i> Legacy template, expand the <b>Legacy Templates</b> tree and double-click on the template name, or right-click and select the <b>Modify Document Template</b> context menu option. The <b>RTF Style Editor</b> displays. |          |
| 4.   | Alternatively, to create a <b>new Legacy template</b> , right-click on <b>Legacy Templates</b> and select the <b>Create RTF Style Template (Legacy)</b> context menu option. Enterprise Architect displays a prompt for the new template name.      |          |
| 5.   | Type the name of the new template and click on the OK button. The <b>RTF Style Editor</b> displays.                                                                                                                                                 |          |

#### **Topics**

| Topic                   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | See also                                                                                                                                 |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RTF Style Editor</b> | <p>The <b>RTF Style Editor</b> contains a list of all available RTF fragments for modification and customization.</p> <p>Each fragment typically contains RTF plus one or more special tag names that Enterprise Architect replaces with information during generation. Currently you cannot alter the content within the tag names, but you can omit a complete tag by removing it, or alter its basic display properties in the surrounding RTF.</p> <p>Special tag names are delimited by # characters; for example, <b>#NOTES#</b></p> <p>Click on the:</p> <ul style="list-style-type: none"> <li>• <b>Get Default</b> button to retrieve the default Enterprise Architect template for the currently-selected template item in the left hand list</li> <li>• <b>Save</b> button to save the version of the template for this</li> </ul> | <a href="#">The Legacy RTF Report Generator</a><br> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See also |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>style only</p> <ul style="list-style-type: none"> <li>• <b>Delete</b> button to remove your modified version of the template, which causes Enterprise Architect to use the default template during report generation</li> </ul> <p>To select a template during report generation, click on the <b>Style</b> drop-down arrow on the <b>Rich Text Format Report</b> dialog. Once a style is selected, Enterprise Architect applies that to the current report. Select &lt;Basic&gt; for the inbuilt style.</p> |          |

### Notes

- The RTF Style Editor discussed here automatically displays when you modify or create a Legacy RTF template. If you select a template created in the enhanced **RTF Style Template Editor**, that editor opens automatically instead
- You can transport these RTF templates between models, using the **Export Reference Data** and **Import Reference Data** options on the **Project | Model Import/Export** menu
- To delete a template, right-click on it and select the **Delete Document Template** context menu option
- You can also **alter the custom language settings**

### Learn more

- [RTF Style Template Editor](#)<sup>[2272]</sup>
- [Export Reference Data](#)<sup>[306]</sup>
- [Import Reference Data](#)<sup>[308]</sup>
- [Custom Language Settings](#)<sup>[2326]</sup>

### 18.1.3.11 Save as Document

The *Document* feature enables a particular documentation configuration to be 'remembered', linking the loaded template within the **Rich Text Format Report** dialog to the current highlighted package. If a particular template is always used with a specific package, and multiple cases of documentation exist to be propagated, saving these as Documents can ease document generation later.

To create and use Documents, follow the steps below.

#### How to

| Step | Action                                                                                                                          | See also                                                          |
|------|---------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| 1.   | Open the <b>Rich Text Format Report</b> dialog (see <b>The Legacy RTF Report Generator</b> for instructions on how to do this). | <a href="#">The Legacy RTF Report Generator</a> <sup>[2315]</sup> |
| 2.   | Click on the <b>Save as Document</b> button. The Save current as document definition dialog displays:                           |                                                                   |

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |          |
| 3.   | In the <b>Enter Value</b> field, type a name for the document and click on the <b>OK</b> button. The document is added to the Resources window for easy future access.                                                                                                                                                                                                                                                                                                                                                                                                              |          |
| 4.   | To generate documentation from the <b>Resources</b> window, right-click on the required document. The context menu displays.                                                                                                                                                                                                                                                                                                                                                                                                                                                        |          |
| 5.   | Select the required option.<br>The context menu options are: <ul style="list-style-type: none"> <li>• <b>Open Document</b> - Opens the corresponding .RTF file, as specified by the RTF template Filename property</li> <li>• <b>Generate Document</b> - Opens the <b>Rich Text Format Report</b> dialog, loaded with the specified template</li> <li>• <b>Auto Generate Document</b> - Generates documentation, with the document located at the path specified by the template's Filename property</li> <li>• <b>Delete Document</b> - Removes the specified document.</li> </ul> |          |


### 18.1.3.12 Custom Language Settings

If you export RTF-format documents from Enterprise Architect in languages other than English, you can customize the codepage, default language ID and character set that Enterprise Architect uses when generating RTF. This makes it much easier to generate documentation appropriate to your country or locale.

You can also set up a list of word substitutions. For instance, where Enterprise Architect would include the word *Figure*, you can specify another word to replace it that is either in your language or more meaningful to your readers.

To set up Substitutions, follow the steps below.

#### How to

| Step | Action                                                                                                                              | See also                                                                                                                              |
|------|-------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| 1.   | Open the <b>Rich Text Format Report</b> dialog (see The Legacy RTF Report Generator for how to do this).                            | <a href="#">The Legacy RTF Report Generator</a>  |
| 2.   | In the <b>Language</b> panel (bottom left of dialog) click on the Adjust button. The <b>Customize RTF Language</b> dialog displays. |                                                                                                                                       |
| 3.   | Double-click on an item to set or clear its <b>Substitute</b> word.                                                                 |                                                                                                                                       |

| Step | Action                                                 | See also |
|------|--------------------------------------------------------|----------|
| 4.   | When you have finished, click on the <b>OK</b> button. |          |

To Set Up Codepage and Character Set, Follow the steps below

#### How to

| Step | Action                                                                                                                                                                              | See also |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | From the drop-down lists in the <b>Language</b> , <b>Codepage</b> and <b>Charset</b> fields, select the language, codepage and character set that most closely match your location. |          |
| 2.   | If required, modify the <b>Substitute Tags</b> by double-clicking on each and manually setting the value (for advanced use only).                                                   |          |
| 3.   | To clear the substitution list, double-click on each item in turn and delete the substitute value.                                                                                  |          |
| 4.   | When you have completed the settings, click on the <b>OK</b> button to save them.                                                                                                   |          |

Now when you generate RTF documents, the substitute tags are used in the output.

#### Notes

- You can transport these language and tag definitions between models, using the **Export Reference Data** and **Import Reference Data** options on the **Project | Model Import/Export** menu

#### Learn more

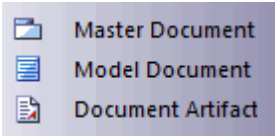

- [Export Reference Data](#)<sup>[306]</sup>
- [Import Reference Data](#)<sup>[308]</sup>

## 18.1.4 Virtual Documents

*Virtual documents* enable you to structure and filter your **RTF** and **HTML** reports by selecting, grouping and ordering individual packages independent of the organization of the Project Browser.

#### Topics

| Topic           | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Abstract</b> | <p>You can create separate virtual documents defining, say, Requirements, Use Cases or Design elements of a project, or you can combine these separate reports - retaining their own different formats - into a single generated document with common headers and footers and a central contents list; this combined document could apply your corporate standards.</p> <p>You generate virtual documents in Enterprise Architect from individual <b>Model Document</b> elements; you can also, if required, combine several Model Documents under a <b>Master Document</b> package element.</p> <p>You can create as many Model Documents as required, for as many combinations of information as required.</p> <p>Each Model Document element has its own template; for example, a specifically-designed Requirements template for a Requirements document, or a Use Case template for a section on Use Cases.</p> <p>For HTML, the template is identified on the report generation dialog; for RTF, the template is identified in a Tagged Value.</p> <p>The content of the report is defined as either:</p> <ul style="list-style-type: none"> <li>• A list of packages (defined as attributes) dragged onto the element in whatever order or combination is most appropriate to your requirements - you can easily <b>add</b> or <b>delete</b> packages as necessary; or</li> <li>• (Not for HTML reports) a standard model search (defined as Tagged Values) created within the Model Search facility - note that diagram searches are not supported; when you generate the document, this search captures the required data throughout the model and populates the document</li> </ul> <p>The Master Document element has its own template Tagged Value, which defines the headers, footers and central contents list; you can <b>import your corporate standards template</b> and edit the Tagged Value to identify that.</p> <p>The template in the Master Document overrides the templates in the Model Documents; for example, headers and footers in the Master Document template override any header and footer definitions in the Model Document templates - this enables you to apply consistent and continuous styles and page numbering throughout the report generated through the Master Document.</p> <p>If you want the Model Documents to have their own styles, applied through their own RTFTemplate Tagged Values, either leave the Master Document RTFTemplate Tagged Value blank (for completely separate overall styles) or remove the definition of specific styles from the Master Document template.</p> <p>You can control the sequence in which information is presented in the document; see the <i>Document Order</i> topic.</p> | <p><a href="#">Create Model Document</a><sup>[2331]</sup></p> <p><a href="#">Create Master Document</a><sup>[2330]</sup></p> <p><a href="#">Add Packages to Model Document</a><sup>[2333]</sup></p> <p><a href="#">Delete Package in Model Document</a><sup>[2334]</sup></p> <p><a href="#">Create and Modify Searches</a><sup>[62]</sup></p> <p><a href="#">Import RTF Template</a><sup>[2316]</sup></p> <p><a href="#">Document Order</a><sup>[2335]</sup></p> |

| Topic                    | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also                                           |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| <b>Document Elements</b> | <p>The Master Document and Model Document elements are available from the Documentation page of the Toolbox; on the Toolbox, select <b>More tools   Documentation</b> (this Toolbox page also provides the <b>Document Artifact</b> element, which is not related to virtual documents but is used for adding a linked document to an element).</p>  <p>When you drag the Master Document and Model Document elements onto a diagram, the following symbols display, respectively:</p>  | <a href="#">Document Artifact</a> <sup>[197]</sup> |

### Notes

- In a Model Document for an RTF report, you should not define both a list of packages and a search; if both are present, when you generate the RTF document Enterprise Architect works from the package list only
- You cannot use **RTF Bookmarking** in Master Document elements, which effectively replace RTF Bookmarking in Word

RTF Bookmarking requires each bookmark to be unique; when you generate a report with a standard RTF template (including in a single Model Document element), each bookmark is unique and there is a 1:1 association between the Elements-details being generated and the elements in the repository

As Master Documents are intended to contain multiple sub-documents, the association ceases to be 1:1; there is no simple method that enables the generated data to be uniquely identified directly in association with the original element

### Learn more

- [Model and Master Documents](#)<sup>[2368]</sup>
- [RTF Bookmarking](#)<sup>[2346]</sup>

### Learning Center topics

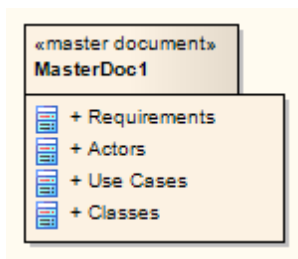
- (Alt+F1) | **Reporting | Virtual Documents**

#### 18.1.4.1 Create Master Document

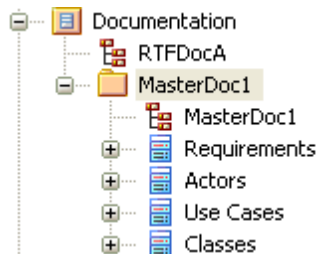
Whilst you can create Model Document elements separately and generate individual documents from each one, you have added flexibility and scope if you organize Model Documents under a Master Document (a package element, with a child Common diagram to which you add the Model Document elements).

You can then generate a document with a corporate template for the covers, contents, headers and footers, whilst each section (generated from a separate Model Document) has its own appearance defined by a template appropriate to the section content.

When you add Model Document elements to a Master Document, the Master Document element resembles the following:



The Master Document element and child diagram display in the Project Browser as shown below:



### How to

To create a Master Document element

| Step | Action                                                                                                                                                       | See also |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | Open or create the diagram (any type, although <i>Documentation</i> automatically provides the correct Toolbox page) in which to create the Master Document. |          |
| 2    | If the Toolbox Documentation page is not already shown, select <b>More tools   Documentation</b> .                                                           |          |
| 3    | Drag the <i>Master Document</i> icon onto the diagram.<br>The system prompts you for the name of the Master Document.                                        |          |

| Step | Action                                                                                                                                                                                                                                                                                                                                                                      | See also                                                      |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| 4    | Type the element name and click on the <b>OK</b> button.<br>The system creates the Master Document element and a child Custom diagram of the same name.                                                                                                                                                                                                                     |                                                               |
| 5    | If creating a Master Document for an HTML report, go to step 7.<br><br>Otherwise, open the Tagged Values window ( <b>View   Tagged Values</b> ) and click on the Master Document element.<br><br>The <i>RTFTemplate</i> Tagged Value displays in the window; this Tagged Value is filled with the valid template names.                                                     |                                                               |
| 6    | The RTFTemplate Tagged Value defaults to ( <i>model document: master template</i> ).<br><br>If you want to use an alternative master template, click on the drop down arrow at the right of the field and click on that template in the list.                                                                                                                               |                                                               |
| 7    | Return to the Project Browser and open the Master Document child diagram.<br>At this point, to provide the content for the generated document, you either: <ul style="list-style-type: none"> <li>• Create the Model Document elements in the child diagram, or</li> <li>• Drag existing Model Document elements from the Project Browser into the child diagram</li> </ul> | <a href="#">Create Model Document</a> <sup>[2331]</sup>       |
| 8    | Having set up the Master Document, and created or added and set up its Model Documents as required, you can generate the RTF report on the Master Document; this pulls in the contents of the Model Documents.                                                                                                                                                              | <a href="#">Generate the Document (RTF)</a> <sup>[2338]</sup> |

**Learn more**

- [Import RTF Template](#)<sup>[2316]</sup>

**Learning Center topics**

- (Alt+F1) | **Reporting | Virtual Documents Create a Master Document**

**18.1.4.2 Create Model Document**

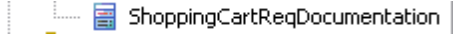
You can create as many Model Document elements as are necessary to provide the sections of your generated document (under a Master Document) or to provide the independent documents you require.

**How to**



To create a Model Document element

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | See also                                                                                                               |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| 1    | <p>Either:</p> <ul style="list-style-type: none"> <li>• Open the child Custom diagram of the Master Document element you have created or</li> <li>• If you are creating independent Model Documents, create a new Documentation diagram (<b>Extended   Documentation</b> in the New Diagram dialog)</li> </ul> <p>The new diagram can live anywhere in the model hierarchy, outside the packages you are adding to the document; you could create a diagram called Documentation within a specific Documentation package, and use this to hold the independent Model Document elements for your virtual documents.</p> <p>You can also create Model Documents on any convenient type of diagram, but you must remember to select the correct Toolbox page, (see step 2).</p> | <p><a href="#">Create Master Document</a><sup>[2330]</sup></p> <p><a href="#">Add New Diagrams</a><sup>[724]</sup></p> |
| 2    | <p>From the Documentation page of the Toolbox (<b>More tools   Documentation</b>) drag the <i>Model Document</i> icon onto the diagram to create a new Model Document element.</p> <p>Give the element an appropriate name: for example, if the documentation is relevant to the shopping cart requirements of a model, you could call it <i>ShoppingCartReqDocumentation</i>.</p> <p>Click on the <b>OK</b> button.</p>                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                        |
| 3    | <p>If you are creating a Model Document element for an HTML report, go now to the <i>Add Packages to Model Document</i> topic.</p> <p>Otherwise, open the Tagged Values window (<b>View   Tagged Values</b>) and click on the Model Document element.</p> <p>The RTFTemplate, SearchName and SearchValue Tagged Values display in the window.</p>                                                                                                                                                                                                                                                                                                                                                                                                                            | <p><a href="#">Add Packages to Model Document</a><sup>[2333]</sup></p>                                                 |
| 4    | <p>Click on the drop-down arrow to the right of the <b>RTFTemplate</b> field, and click on the template to use for this Model Document.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                        |
| 5    | <p>If you are creating a list of packages for the Model Document, go now to the <i>Add Packages to Model Document</i> topic.</p> <p>Otherwise, click on the drop-down arrow to the right of the <b>SearchName</b> field, and click on the model search type to populate this Model Document.</p> <p>Remember that, in a Model Document for an RTF report, you should not define both a list of packages and a search; if both are present, when you generate the RTF document Enterprise Architect works from the package list only.</p>                                                                                                                                                                                                                                     | <p><a href="#">Add Packages to Model Document</a><sup>[2333]</sup></p>                                                 |

| Step | Action                                                                                                                                                                                                                                                                                                                                                        | See also                                         |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| 6    | If necessary, type a search term in the <b>SearchValue</b> field.                                                                                                                                                                                                                                                                                             |                                                  |
| 7    | <p>Create further Model Document elements as required.</p> <p>Your Model Document element appears in the Project Browser with a Class icon, as shown below:</p>  <p>When you have created all the required Model Document elements, see the <i>Document Order</i> topic.</p> | <a href="#">Document Order</a> <sup>[2335]</sup> |

#### Notes

- Diagram Searches are not supported
- Custom SQL searches are supported if they are returning elements; the SQL must include *ea\_guid AS CLASSGUID* and the *object type*

#### Learn more

- [Creating Search Definitions](#)<sup>[623]</sup>

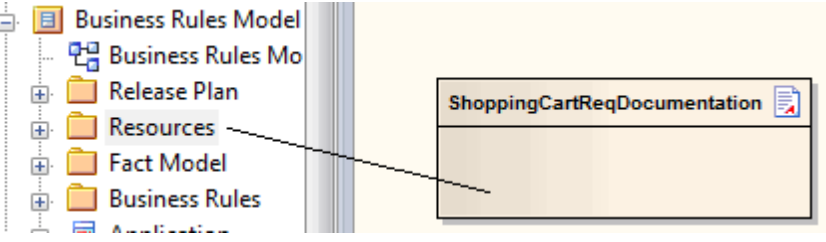
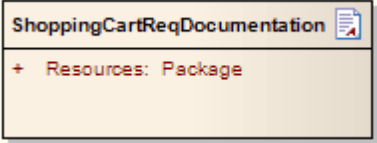
### 18.1.4.3 Add Packages to Model Document

#### How to

To add packages to your Model Document element, follow the steps below.

(As the example Model Document element here is called *ShoppingCartDocumentation*, the steps indicate how to add shopping cart-related packages to the element.)

| Step | Action                                                                                                                                                         | See also |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | Keeping the documentation diagram open, find a package in the Project Browser to add to the documentation; for example, a Resources package in a Dynamic view. |          |
| 2    | Drag and drop the package from the Project Browser onto the Model Document element, as shown below:                                                            |          |

| Step | Action                                                                                                                                                                                                                                                                                                        | See also                                                                                                                                                                                                                                                                              |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|      |                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                       |
| 3    | <p>The title of the package displays in the Model Document element in the <i>Attributes</i> compartment, as shown below:</p>  <p>This means that the Resources package is included in the document when you generate it.</p> |                                                                                                                                                                                                                                                                                       |
| 4    | <p>Using the above method, you can add as many packages from as many different views as required.</p> <p>The next step is to generate your document in RTF or HTML, but consider the impact of your package list on the Document Order; you can also delete packages if required.</p>                         | <p><a href="#">Generating RTF Documents</a> <sup>[2338]</sup></p> <p><a href="#">Generating a Master Document - HTML</a> <sup>[2368]</sup></p> <p><a href="#">Document Order</a> <sup>[2335]</sup></p> <p><a href="#">Delete Packages from a Model Document</a> <sup>[2334]</sup></p> |

### Learning Center topics

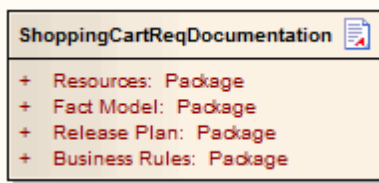
- (Alt+F1) | **Reporting | Virtual Documents | Create from Packages**

#### **18.1.4.4 Delete Package in Model Document**

You can delete a package from your Model Document element.

This example includes four packages:

- Resources Package
- Activity Diagram Package
- View Cart Package
- Sequence Diagram Package



### How to

To delete a package from a Model Document element

| Step | Action                                                                                            | See also |
|------|---------------------------------------------------------------------------------------------------|----------|
| 1    | In the Project Browser, expand the element to list the package attributes.                        |          |
| 2    | Right-click on the package to delete, and select the <b>Delete Attribute</b> context menu option. |          |

Alternatively:

| Step | Action                                                                                                                                                                           | See also |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | In either the Project Browser or the diagram, right-click on the Model Document element and select the <b>Attributes</b> context menu option.<br>The Attributes dialog displays. |          |
| 2    | On the <b>Attributes</b> list, click on the package to delete.                                                                                                                   |          |
| 3    | Click on the <b>Delete</b> button to remove the package from the document element.                                                                                               |          |

### Learning Center topics

- (Alt+F1) | [Reporting | Virtual Documents | Remove a Package](#)

#### **18.1.4.5 Document Order**

The order in which information is compiled into a document depends on:

- The sequence of Model Document elements in a Master Document element
- Whether you define a Model Search in a Model Document element (not for HTML reports)
- Whether you define a package list in a Model Document element

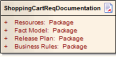
When you have considered and, if necessary, amended the order in which information is compiled, you can generate the document in RTF or HTML.

### Topics

| Topic                          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See also |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Model Document Sequence</b> | <p>When you generate a document from a Master Document element, the sequence in which the sections are generated is determined by the order in which the child Model Document elements are listed in the Project Browser.</p> <p>You can create elements anywhere in a diagram, therefore the generator refers to the Project Browser sequence.</p> <p>If necessary, change the sequence using the green Up and Down arrows in the Project Browser toolbar to move an element up or down within the package.</p> |          |
| <b>Model Search</b>            | <p>A Model Search operates on the database, and accesses records in the order in which they are stored; this order depends on many factors, and can change with database maintenance.</p> <p>Therefore, the sequence of information provided by the search is unpredictable.</p>                                                                                                                                                                                                                                 |          |
| <b>Package Order</b>           | <p>When you create a package list in a Model Document element, the sequence of information is determined by the order in which the package attributes are listed within the element.</p> <p>You can change the sequence using the Attributes dialog, and if you prefer a package to be in a different section of the document, you can move the attribute from one Model Document element to another.</p> <p>Both these procedures are described below.</p>                                                      |          |

### How to

To change the sequence of Packages in a Model Document element, refer to the example element and follow the steps below:

| Images                                                                              | Step | Action                                                                                                                                                                                   | See also |
|-------------------------------------------------------------------------------------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|  | 1    | <p>Right-click on the Model Document and select the <b>Attributes</b> option from the context menu.</p> <p>The Attributes dialog displays.</p>                                           |          |
|                                                                                     | 2    | <p>On the <b>Attributes</b> list, click on a package to move and click on the Up or Down (hand) buttons to change the order in which the packages are included in the documentation.</p> |          |

| Images | Step | Action                                                                                    | See also |
|--------|------|-------------------------------------------------------------------------------------------|----------|
|        | 3    | When you are satisfied with the order of your packages, click on the <b>Close</b> button. |          |

To move a package attribute from one Model Document to another, follow the steps below

| Step | Action                                                                                                                                              | See also |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | Expand the Model Document elements in the Project Browser, so that both list their package attributes.                                              |          |
| 2    | Click-and-hold on the attribute to move, and drag it onto the name of the target Model Document element.                                            |          |
| 3    | Release the mouse button.<br>The attribute is removed from the source element and added to the top of the list of attributes in the target element. |          |
| 4    | If necessary, move the attribute down the attribute list, as described in <i>To change the sequence of Packages</i> , above.                        |          |

#### Learn more

- [Generate the Document \(RTF\)](#)  <sup>[2338]</sup>
- [Generate a Master Document HTML](#)  <sup>[2368]</sup>

#### Learning Center topics

- **(Alt+F1) | Reporting | Virtual Documents | Reorder Packages**

### 18.1.4.6 Section Numbering in Virtual Documents

Virtual documents can contain several sections based on separate templates. To ensure that section and subsection numbering continues sequentially through the sections, you can either use:

- The *MasterList* style from the **Normal.rtf** file (which is the default applied to all new templates, such as those used in each Master Document and Model Document element); this provides a simple but consistent list numbering style that you can start off with. Alternatively, use:
- Your own numbering style

If you design your own styles in the template used in each model document, each template must use the same **List Overrides** for section numbering. These have a unique ID, so to propagate the List Overrides across multiple templates, you must:

| Step | Action                                                                                                                                                                                       | See also                                                 |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| 1    | Set up the common List Overrides in the <i>Normal.rtf</i> template (your own customized override list).                                                                                      |                                                          |
| 2    | Within each of the other templates used in the virtual document, using the template editor, select <b>File   Update Styles</b> to update the templates to include the common List Overrides. | <a href="#">File and Print Options</a> <sup>[2277]</sup> |
| 3    | In each template use this common List Level for progressive numbering throughout the virtual document.                                                                                       |                                                          |

#### Learn more

- [RTF Templates](#) <sup>[2270]</sup>
- [User Defined Section Numbering](#) <sup>[2299]</sup>

### 18.1.4.7 Generate the Document (RTF)

#### How to

To generate the documentation defined in the Master Document and/or Model Documents

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                    | See also                                                            |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| 1    | On the documentation diagram, click on the Master Document element (or on an independent Model Document element) and press <b>(F8)</b> .<br>The Generate RTF Documentation dialog displays.                                                                                                                                                                                                               |                                                                     |
| 2    | Set the options for your RTF document as required.<br>See <i>Generate RTF Documentation Dialog</i> and related topics for further information on these settings.                                                                                                                                                                                                                                          | <a href="#">Generate RTF Documentation Dialog</a> <sup>[2256]</sup> |
| 3    | Click on the <b>Generate</b> button to create the documentation.<br>The RTF Report Generator works through the defined content of the Master Document element and/or the Model Document elements and pulls in the information from either the listed packages or the executed searches, formatted according to the templates identified in the <i>RTFTemplate</i> Tagged Value for each document element. |                                                                     |
| 4    | If you have not selected the <b>View Document on Completion</b> checkbox, click on the <b>View</b> button to view the documentation.                                                                                                                                                                                                                                                                      |                                                                     |

### 18.1.5 Other Documents

Enterprise Architect has other RTF based documentation that you can output:

| Topic                                                                                                                                                                                               | Link                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| <b>Dependency Report</b> - shows a list of any elements that are dependent (in a Dependency connector) on another element for their specification.                                                  | <a href="#">Dependency Report</a> <sup>[2339]</sup>      |
| <b>Diagrams Only Report</b> - lists only the relevant diagrams from the target package.                                                                                                             | <a href="#">Diagrams Only Report</a> <sup>[2340]</sup>   |
| <b>Implementation Report</b> - lists, for a specified package, the elements that require implementers, together with any source elements in Realize (Implements) relationships with those elements. | <a href="#">Implementation Report</a> <sup>[2341]</sup>  |
| <b>Resource Report</b> - shows how your resources are deployed in your project, displaying a list of all elements that have resources allocated to them.                                            | <a href="#">Resource Report</a> <sup>[477]</sup>         |
| <b>Testing Report</b> - outputs the test scripts and results you have entered against elements in the model, in Rich Text format.                                                                   | <a href="#">Testing Report</a> <sup>[2342]</sup>         |
| <b>Testing Details Report</b> - enables you to review the testing details for a package, as filtered reports on all elements in the package hierarchy under the selected package .                  | <a href="#">Testing Details Report</a> <sup>[2343]</sup> |

#### 18.1.5.1 Dependency Report

A Dependency report shows a list of any elements in the package (selected from the Project Browser) that are dependent on another element for their specification.

For example, a Use Case derives its specification from the Requirement that it realizes.

Each of the elements in the first column of the report is the source or dependent in a Dependency connector to the corresponding target element in the **Dependent on** column.


Access **Project | Documentation | Dependency Details**

#### How to

On the Dependency Details dialog, run the report and make selections from the options as indicated below:

| Field               | Usage                                                                                        | See also |
|---------------------|----------------------------------------------------------------------------------------------|----------|
| <b>Root Package</b> | Confirm the root package; all elements and packages under this package appear in the report. |          |



| Field                | Usage                                                                                                                              | See also                                                                                                          |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
|                      | If you want to change the package, click on the replacement package in the Project Browser and click on the <b>Refresh</b> button. |                                                                                                                   |
| <b>Refresh</b>       | Run the report.                                                                                                                    |                                                                                                                   |
| <b>Locate Object</b> | Locate the selected element in the Project Browser.                                                                                |                                                                                                                   |
| <b>Print</b>         | Print the list.                                                                                                                    |                                                                                                                   |
| <b>Save Report</b>   | Display a small dialog in which you specify the file location into which to save the report as an RTF file.                        |                                                                                                                   |
| <b>Details</b>       | List dependency details; lists the elements in the current package and the elements that they are dependent on.                    | <a href="#">Dependency</a><br> |

### 18.1.5.2 Diagrams Only Report

You can also produce an RTF report that contains only the relevant diagrams from the target package. This is convenient for printing or handling a lot of diagrams in batch, rather than exporting or printing each one at a time.

**Access** [Project Browser package context menu | Documentation | Diagrams Only Report](#)

#### How to

To produce a Diagrams Only report

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See also |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | On the Export Diagrams to RTF Document dialog, in the <b>Output Path</b> field, type or browse for the output location to create the report in.                                                                                                                                                                                                                                                                                                                                                                                                                                      |          |
| 2    | Select the options you require, as follows: <ul style="list-style-type: none"> <li>• Select the <b>Embed Diagrams in Document</b> checkbox to ensure the diagrams are created within the RTF file, not as linked image files</li> <li>• Select the <b>Include all child packages</b> checkbox to document all of the diagrams included in any child package</li> <li>• Select the <b>Include Diagram Name</b> checkbox to include the diagram name within the generated documentation</li> <li>• Select the <b>Order Diagrams Alphabetically</b> checkbox to generate the</li> </ul> |          |

| Step | Action                                                                                | See also |
|------|---------------------------------------------------------------------------------------|----------|
|      | documentation in alphabetical order                                                   |          |
| 3    | Click on the <b>Generate</b> button to run the report.                                |          |
| 4    | When the report is generated, click on the <b>View</b> button to show the RTF output. |          |

### 18.1.5.3 Implementation Report

An *Implementation report* shows a list of the elements in the package (selected in the Project Browser) that require implementers, together with any source elements in Realize (Implements) relationships with those elements.

**Access** [Project | Documentation | Implementation Details](#)

#### How to

On the Implementation Details dialog, make selections from the options as indicated below, and run the report:

| Field                     | Usage                                                                                                                                                                                                                                                            | See also                                                        |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
| <b>Root Package</b>       | Confirm the root package; all elements and packages under this package appear in the report.<br><br>If you want to report on a different package, click on that package in the Project Browser.                                                                  |                                                                 |
| <b>Set Target Types</b>   | By default Enterprise Architect only reports on a limited number of element types, such as Use Cases and Requirements; you can expand the range of element types to include.<br><br>Set the list of types to report on, using the Implementation Targets dialog. | <a href="#">Implementation Targets Dialog</a> <sup>[2342]</sup> |
| <b>Refresh List</b>       | Run the report and display the results.                                                                                                                                                                                                                          |                                                                 |
| <b>Locate Object</b>      | Locate the selected element in the Project Browser.                                                                                                                                                                                                              |                                                                 |
| <b>Show Unimplemented</b> | Show non-implemented elements.<br><br>Non-implemented elements are those that don't have any other element to realize them (for example, a Use Case that has no Component or Class to implement the Use Case behavior).                                          |                                                                 |

| Field                   | Usage                                                                                                                                                                                           | See also                                  |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
|                         |                                                                                                                                                                                                 |                                           |
| <b>Show Implemented</b> | Show implemented elements.<br><br>These are elements that do have some element associated with them in a Realize relationship (for example a Use Case that has a Component that implements it). | <a href="#">Realize</a> <sup>[1254]</sup> |
| <b>Details</b>          | List elements in the current package hierarchy and elements that implement them.                                                                                                                |                                           |
| <b>Print</b>            | Print the list.                                                                                                                                                                                 |                                           |
| <b>Save Report</b>      | Specify the file location into which to save the report as an RTF file.                                                                                                                         |                                           |

#### 18.1.5.3.1 Implementation Targets Dialog

The Implementation Targets dialog is accessed by clicking on the **Set Target Types** button on the Implementation Details dialog.

This dialog enables you to set the types of elements that appear in the report as requiring implementation. Double-click on an element in either list to move it to the other list.

#### 18.1.5.4 Testing Report

Enterprise Architect enables you to output the test scripts and results you have entered against elements in the model, in Rich Text format.

**Access** [Project | Documentation | Testing Report](#)

##### How to

To view a Testing report

| Step | Action                                                                                                                        | See also |
|------|-------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | In the Project Browser, click on the package to report on.<br><br>The package name displays in the <b>Root Package</b> field. |          |
| 2    | In the Report on panel, select the check box against each type of test to include in the report.                              |          |
| 3    | If you intend to report on tests on elements in the subordinate packages, select the <b>Include child packages</b> checkbox.  |          |

| Step | Action                                                                                                | See also |
|------|-------------------------------------------------------------------------------------------------------|----------|
|      |                                                                                                       |          |
| 4    | In the <b>Output file</b> field, type in or browse for the file path of the report output file.       |          |
| 5    | Click on the <b>Generate</b> button to output the report, and open it at the specified file location. |          |

### 18.1.5.5 Testing Details Report

The Testing Details report enables you to review the testing details for a package, running filtered reports on all elements in the package hierarchy under the selected package.

**Access** [Project | Documentation | Testing Details](#)

#### Use to

- Show the testing details for the elements in a package, for tests that meet specific criteria

#### How to

On the Testing tab, make selections from the options as indicated below, and run the report:

| Field                | Usage                                                                                                      | See also |
|----------------------|------------------------------------------------------------------------------------------------------------|----------|
| <b>Run By</b>        | Select a name to filter for tests run by that person; click on the <b>x</b> button to clear the field.     |          |
| <b>Checked By</b>    | Select a name to filter for tests checked by that person. Click on the <b>x</b> button to clear the field. |          |
| <b>Test Type</b>     | Select the radio button for the required test type.                                                        |          |
| <b>Status</b>        | Select the radio button for the required test status.                                                      |          |
| <b>Locate Object</b> | (After clicking on an element in the Test Details list) locate the element in the Project Browser.         |          |
| <b>Refresh</b>       | Re-run the report query.                                                                                   |          |
| <b>Print</b>         | Print a summary of the test results.                                                                       |          |

**Notes**

- If required, you can display the test details on the Testing window, by double-clicking on the test line in the report

**Learn more**

- [The Testing Workspace](#)<sup>[2213]</sup>

**18.1.6 Use MS Word**

When Enterprise Architect only supported the 'Legacy' RTF Report Generator, MS Word provided many additional features in RTF report generation.

However, as the 'enhanced' RTF Report Generator has been developed within Enterprise Architect, the facilities provided by Word have been incorporated into the RTF Generator or otherwise rendered unnecessary. It is therefore likely that you would no longer have any requirement to use Word in generating your RTF reports.

These topics on the use of MS Word are therefore included only to support users who are still working with the 'Legacy' RTF Report Generator, or who are using Word for personal preference.

**Topics**

| Topic        | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See also                                                                                                                                                                                                                                                                                                                        |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Usage</b> | <p>To further enhance and customize RTF documentation it is possible to create a custom master document, which can be used to add a table of contents, table of figures, headers and footers and to refresh linked files.</p> <p>In addition it is possible to create documents with sustainable links to generated 'pieces' of Enterprise Architect output, pre-divided by Enterprise Architect using <b>bookmarks</b>.</p> <p>As an alternative to the <i>Word</i> master document and to RTF Bookmarking, internal to Enterprise Architect, you can create <i>Virtual Documents</i>.</p> <p>In addition to creating Word master documents, you can:</p> <ul style="list-style-type: none"> <li>• <b>Open a Report in Microsoft Word</b></li> <li>• <b>Change Linked Images to Embedded</b></li> <li>• <b>Apply Other Features of Word.</b></li> </ul> | <p><a href="#">RTF Bookmarks</a><sup>[2346]</sup></p> <p><a href="#">Virtual Documents</a><sup>[2327]</sup></p> <p><a href="#">Open a Report in Microsoft Word</a><sup>[2345]</sup></p> <p><a href="#">Change Linked Images to Embedded</a><sup>[2345]</sup></p> <p><a href="#">Other Features of Word</a><sup>[2347]</sup></p> |

**Notes**

- You can develop a report using the combined facilities of Word and Enterprise Architect with few problems, as long as you leave definition of the section styles to the final stages in Enterprise Architect just prior to report generation

Word truncates the section bookmarks, as it uses a smaller field length for sections

In Word, you can review and edit reports generated by Enterprise Architect, but you cannot import them back into Enterprise Architect without damaging the section style definition

### 18.1.6.1 Open a Report in Microsoft Word

#### Topics

| Topic        | Detail                                                                                                                                                                                                                               | See also |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Usage</b> | To open an RTF file in MS Word, simply load Word and open the file as a normal document.<br><br>Word converts the file; if Word is the default handler of RTF files, double-click on the output file to load up and view the report. |          |

#### Notes

- If you have Word configured to view RTF files, you can also click on the **View Output** button on the Generate RTF Documentation dialog

### 18.1.6.2 Change Linked Images to Embedded

One of the options available when generating RTF documentation is the ability to store image files in a separate directory to the RTF document. If at a later stage it becomes desirable to embed the images in the RTF documentation, this is especially important when the document is to be distributed. If the images are stored in a separate directory recipients of the document see only the placeholder of images rather than the actual images.

If you import an RTF document into Word with the images not embedded into the document, you have the option of breaking the links to the images and saving the image in the document.

To break image links in Word, follow the steps below

#### How to

| Step     | Action                                                        | See also |
|----------|---------------------------------------------------------------|----------|
| <b>1</b> | Open the required RTF file in Word.                           |          |
| <b>2</b> | Select the <b>Edit   Links</b> menu option.                   |          |
| <b>3</b> | Highlight all links in the <b>Links</b> list.                 |          |
| <b>4</b> | Select the <b>Save Picture in Document</b> checkbox.          |          |
| <b>5</b> | Click on the <b>Break Link</b> button.                        |          |
| <b>6</b> | When prompted, click on the <b>Yes</b> button to break links. |          |

| Step | Action | See also |
|------|--------|----------|
|      |        |          |

Word breaks the links and saves copies of the images inside the document. You can distribute this document without the image directory.

### 18.1.6.3 RTF Bookmarks

Bookmarks are markers that are automatically placed in your rich text document when you generate it. You can create a master document in Word and link to sections of an Enterprise Architect report based on bookmarks. For example, a Word document might have a section for a small part of your component model. Using bookmarks you can generate a full component model, and then link into just one section of the report.

This way you can maintain a complex Word document from parts of Enterprise Architect reports. If you link into Enterprise Architect reports, then you can regenerate the report and refresh Word links to update the master document without having manually changed anything. For more information on refreshing links, see the **Refresh Links** topic.

Bookmarks are GUID-based numbers that can be created for packages, diagrams and elements. A package bookmark applies from the beginning of a package to the end, and includes all child packages and elements underneath.

To bookmark a Section of Enterprise Architect for RTF Documentation, follow the steps below

#### How to

| Step | Action                                                                                                                                      | See also |
|------|---------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | In the Enterprise Architect <b>Project Browser</b> , right-click on the package to include in the documentation. The context menu displays. |          |
| 2.   | Select the <b>Documentation   Copy RTF Bookmark</b> menu option to paste the package into the clipboard as a bookmark for use in Word.      |          |

To Insert a Bookmarked Section of an Enterprise Architect RTF Document into Word, follow the steps below

#### How to

| Step | Action                                                                                   | See also |
|------|------------------------------------------------------------------------------------------|----------|
| 1.   | Open the Word document and position the cursor at the point at which to insert the file. |          |
| 2.   | Select the Word <b>Insert   File</b> menu option. The <b>Insert</b> dialog displays.     |          |
| 3.   | Locate and click on the file to insert, then click on the <b>Range</b> button.           |          |

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See also                                               |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| 4.   | In the <b>Range</b> cell type or paste the information from the clipboard.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                        |
| 5.   | Click on the <b>OK</b> button.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                        |
| 6.   | <p>Click on the drop-down arrow next to the <b>Insert</b> button. Select the <b>Insert as Link</b> option.</p> <p>The <b>Insert</b> option sets a permanent copy; the <b>Insert as Link</b> option creates a link that is updateable on altering the source document. For <b>Insert as Link</b> to operate you must first set <b>Refresh Links</b>.</p> <p>Every package is bookmarked in the RTF document according to the following rules:</p> <ul style="list-style-type: none"> <li>• All alphabetic and numeric characters remain the same</li> <li>• All other characters (including spaces) are converted to underscores</li> </ul> <p>For example UC01: Use Case Model becomes UC01__Use_Case_Model.</p> | <a href="#">Refresh Links</a><br><small>[2351]</small> |

### Notes

- You cannot use RTF Bookmarking in **Master Document** elements, which effectively replace RTF Bookmarking in Word.

RTF Bookmarking requires each bookmark to be unique. When you generate a report with a standard RTF template (including in a single Model Document element), each bookmark is unique and there is a 1:1 association between the Elements-details being generated and the elements in the repository. As Master Documents are intended to contain multiple sub-documents, the association ceases to be 1:1. There is no simple method that enables the generated data to be uniquely identified directly in association with the original element

### Learn more





- [Refresh Links](#)[2351]
- [Create a Master Document](#)[2330]

#### 18.1.6.4 Other Features of Word

Word offers a considerable number of document enhancement tools to complete your project documentation. Here are some of the things you can do with Word in Enterprise Architect generated RTF documentation:

| Topic                   | Link                                                          |
|-------------------------|---------------------------------------------------------------|
| Add a Table of Contents | <a href="#">Add a Table of Contents</a> <small>[2348]</small> |



| Topic                       | Link                                                                                                                            |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Add a Table of Figures      | <a href="#">Add a Table of Figures</a>       |
| Add Headers and Footers     | <a href="#">Add Headers and Footers</a>      |
| Manipulating Tables in Word | <a href="#">Manipulating Tables in Word</a>  |
| Refresh Linked Files        | <a href="#">Refresh Linked Files</a>         |

### Notes

- Enterprise Architect provides the basic content for your document - use Word to add the presentation and linkages

#### 18.1.6.4.1 Add Table of Contents

Among the features of MS word that can be incorporated into generated Enterprise Architect reports is the option to include a table of contents. A table of contents can be used to aid navigation of documentation and enhance the readability of Enterprise Architect RTF reports. This option provides hyperlinks to the diagrams included in the RTF documentation in the electronic version, and page numbers for both the printed and electronic documentation.

To include a Table of Contents in the RTF documentation, follow the steps below.

### How to

| Step | Action                                                                                                         | See also |
|------|----------------------------------------------------------------------------------------------------------------|----------|
| 1    | Open the Enterprise Architect RTF report to which to add a Table of Contents in MS Word.                       |          |
| 2    | Select the <b>Insert   Reference   Index and Tables</b> menu option.                                           |          |
| 3    | Click on the Table of Contents tab to set the options that are available for formatting the table of contents. |          |

The format of the table of contents is dependant on the heading levels created when the RTF is generated. To set the heading styles in Enterprise Architect RTF documentation, see the *RTF Document Options* topic.

### Learn more

- [Document Options](#) 

#### 18.1.6.4.2 Add Table of Figures

Among the features of MS word that can be incorporated into generated Enterprise Architect reports is the option to include a table of figures. A table of figures can be used to aid the navigation of the documentation and enhance the readability of Enterprise Architect RTF reports. This option provide hyperlinks to the diagrams included in the RTF documentation in the electronic version and page numbers for both the printed and electronic documentation.

To include a Table of Figures in the RTF documentation, follow the steps below

##### How to

| Step | Action                                                                                                              | See also |
|------|---------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | Open the Enterprise Architect RTF report to which to add a Table of Figures in MS Word.                             |          |
| 2.   | Select the <b>Insert   Reference   Index and Tables</b> menu option.                                                |          |
| 3.   | Click on the <b>Table of Figures</b> tab to set the options that are available for formatting the table of figures. |          |

#### 18.1.6.4.3 Add Headers and Footers

Among the features of MS word that can be used to enhance the appearance of Enterprise Architect RTF reports is the ability to add headers and footers to the documentation.

To include headers and footers in the RTF documentation follow the steps below

##### How to

| Step | Action                                                                                    | See also |
|------|-------------------------------------------------------------------------------------------|----------|
| 1.   | In MS Word, open the Enterprise Architect RTF report to which to add headers and footers. |          |
| 2.   | Select the <b>View   Header and Footer</b> menu option.                                   |          |

This enables you to enter information into the header section and the footer section of the RTF Documentation.

#### 18.1.6.4.4 Manipulate Tables in Word

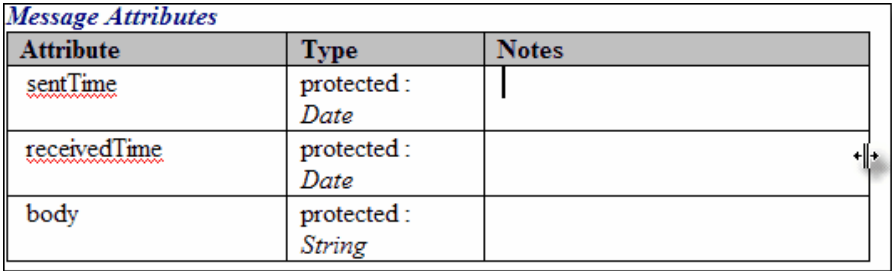
When generating RTF documentation from Enterprise Architect, tables are included when items such as **Attributes** and **Methods** are selected in the **For each Object** section in the **Rich Text Format Report** dialog. MS Word offers several levels of customization for tables and can be used to tidy the formatting of the tables in situations where the margins of the table exceed the dimensions of the page size selected in Word for printing.

**Resize Tables:**

When the amount of detail for a documented item such as an attribute or operation exceeds the margins of the page in MS Word it is necessary to manually resize the table in order to view all of the details.

To manually resize the table follow the steps below

**How to**

| Step | Action                                                                                                                                                                                   | See also |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | Select the table that exceeds the margin size.                                                                                                                                           |          |
| 2    | <p>Mouse over the border of the table until the mouse pointer changes into the icon shown below.</p>  |          |
| 3    | Drag the cursor to the left to reduce the width of the table and then select the <b>File   Print Preview</b> menu option to confirm that the table borders are within the page margins.  |          |
| 4    | Resize all of the tables that overhang the margins of the page by using the steps detailed above.                                                                                        |          |

**Applying Styles to Tables:**

One of the customizable properties of MS Word when working with tables is the ability to apply a style to a table, which enables you to rapidly change the appearance of the table.

To apply styles to a table, follow the steps below

**How to**

| Step | Action                                                                        | See also |
|------|-------------------------------------------------------------------------------|----------|
| 1    | Open the Enterprise Architect RTF report in which to change the table styles. |          |

| Step | Action                                                                                         | See also |
|------|------------------------------------------------------------------------------------------------|----------|
| 2    | Locate and select the table for which to adjust the appearance.                                |          |
| 3    | Select the <b>Table   Table Auto Format</b> menu option. The Table Autoformat dialog displays. |          |

From here you can specify a predefined table style from the **Table styles** list, or create a new style by clicking on the **New** button. The table styles defined in the Table Autoformat dialog only apply to one table at a time so you must apply the style to each table created individually.

#### 18.1.6.4.5 Refresh Links

If you link into Enterprise Architect reports, then you can regenerate the report and refresh MS Word links to update the Word master document without having manually changed anything.

To ensure that links are refreshed in the master document, you must select the **Update automatic links at Open** checkbox in Word.

To ensure that this setting is established follow the steps below.

##### How to

| Step | Action                                                                                       | See also |
|------|----------------------------------------------------------------------------------------------|----------|
| 1.   | From within MS Word select the <b>Tools   Options</b> menu option.                           |          |
| 2.   | Select the <b>General</b> tab and select the <b>Update automatic links at Open</b> checkbox. |          |

## 18.2 HTML Reports



### Topics

| Topic           | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See also                                                |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| <b>Abstract</b> | <p>Enterprise Architect provides automated web-based publishing of models. A new outline structure closely mirrors the model hierarchy and makes it very simple to explore models on-line. With a great new look and the ability to explore very large models efficiently on-line, the new web-publishing capability is a significant enhancement.</p> <p>Enterprise Architect enables export of an entire model or a single branch of the model to HTML Web pages. The <b>HTML report</b> provides an easy to use, highly detailed, Javascript based model tree. In addition, hyperlinked elements make browsing to related information very simple.</p> <p>The current implementation is based on internal and external templates and generated Javascript. The ability to edit all templates is to be added in a future version of Enterprise Architect.</p> | <a href="#">Create an HTML report</a> <sup>[2353]</sup> |

### Notes

- You can create **Web Style Templates** to customize your HTML output
- You can use Master Documents and Model Documents to select specific areas of your model for reporting
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Generate Documents** permission to generate HTML documents

### Learn more

- [Web Style Templates](#)<sup>[2355]</sup>
- [Permission List](#)<sup>[264]</sup>

### Learning Center topics

- (Alt+F1) | **Reporting | HTML Reports**

## 18.2.1 Create an HTML Report

You create an HTML report on a selected package in the Project Browser. All child packages of the selected package are automatically included in the report.

**Access** [Project Browser package context menu | Documentation | HTML Report](#)

### Reference

You create an HTML report using the Generate HTML Report dialog, as indicated below:

| Field/ Button    | Action                                                                                                       | See also |
|------------------|--------------------------------------------------------------------------------------------------------------|----------|
| <b>Output to</b> | Select an output directory for your report.<br>Set any other required options.                               |          |
| <b>Generate</b>  | Click on this button to generate the report.                                                                 |          |
| <b>Progress</b>  | Shows the accumulating percentage completion of the report generation process.                               |          |
| <b>View</b>      | Once the report is complete, click on this button to launch your default HTML viewer and view the web pages. |          |

### Topics

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See also                                                                                                                                                 |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Compatibility</b> | <p>The web report produced is compatible with any standard web server, on either Unix or Windows platforms. Simply bundle up the entire output directory and place it within the context of your web server. All path names should be relative and case sensitive.</p> <p>When you view the HTML report in your HTML viewer, you can switch directly to a page for a specific diagram or element by specifying the appropriate GUID after the report web address. That is:</p> <pre>ht t p : // pat h / pat h / pat h / I ndex . ht m ? gui d = x x x x x x x x x x x x</pre> <p>The word <i>guid</i> must be in lower case, and the value must not include braces { }.</p> <pre>ht t p : // . . . / pat h / I ndex . ht m ? gui d = DC62B0DA- 0D60- 4447- 85E6- B9BBAE7FC90F</pre> <p>To obtain the GUID, right-click on the diagram, package or element in the Project Browser and select the <b>Copy Reference</b> context menu option.</p> <p>The page locator does not work directly in Internet Explorer. Firefox automatically</p> | <p><a href="#">Diagram Menu</a><sup>[58]</sup></p> <p><a href="#">Element Menu</a><sup>[57]</sup></p> <p><a href="#">Package Menu</a><sup>[57]</sup></p> |

| Topic              | Detail                                                                                                                                                                                                                                           | See also |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                    | converts the path to <b>file:///C:/path</b> protocol, and actions it. That protocol also works in Internet Explorer. Therefore, to use the absolute references without a web server you must access the path using the <b>file:///</b> protocol. |          |
| <b>Quick Start</b> | To generate an HTML report right now, follow the steps above on the <i>System Model</i> package of the <i>EAExample</i> project.                                                                                                                 |          |

### Notes

- If you are using Microsoft Internet Explorer 7.0 or later, and you do not have it open, its security profile might block the report display; click on the explanation banner at the top of the screen and select the **Allow Blocked Content** context menu option

### Learn more

- [Generate HTML Report dialog](#) <sup>2354</sup>

### Learning Center topics

- (Alt+F1) | **Reporting | HTML Reports**

## 18.2.2 The Generate HTML Report Dialog

The **Generate HTML Report** dialog is used to generate documentation about your model in HTML format. There are various settings to choose from to control the output, as described below.

| Field               | Usage                                                                                                                             | See also                                            |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| <b>Package</b>      | Display the name of the package you are creating documentation for.                                                               |                                                     |
| <b>Title</b>        | Type the title for your HTML documentation; defaults to <b>Package</b> .                                                          |                                                     |
| <b>Output to</b>    | Type the directory path your documentation is saved to.                                                                           |                                                     |
| <b>Style</b>        | Select a web style template to apply to your documentation (optional).                                                            | <a href="#">Web Style Templates</a> <sup>2355</sup> |
| <b>File</b>         | Specify the file extension for your HTML documentation files; the default is <b>.htm</b> .                                        |                                                     |
| <b>Header Image</b> | Enter or select the file path for the header image graphic to display on your HTML output. If you do not specify a file path, the |                                                     |

| Field                                  | Usage                                                                                    | See also |
|----------------------------------------|------------------------------------------------------------------------------------------|----------|
|                                        | image defaults to the Enterprise Architect logo.                                         |          |
| <b>Preserve White space in Notes</b>   | Preserve existing white space in your notes; deselect to remove white space.             |          |
| <b>No page for Note and Text items</b> | Omit the page for your notes and text items in the HTML report.                          |          |
| <b>Default Diagram</b>                 | Select the diagram the report should open to when the generated documentation is loaded. |          |
| <b>Image Format</b>                    | Select the image file format to save your images in, either PNG or GIF.                  |          |
| <b>Include</b>                         | Select each area of your model to include in your report.                                |          |
| <b>System</b>                          | Select each section to generate in your report.                                          |          |

Click on the **Generate** button to generate the HTML report with the settings you have defined.

Click on the **View** button to display the report you have generated.

#### Notes

- You can also select packages in your model to exclude them (and their subordinate packages) from the HTML reports you generate

#### Learn more

- [Exclude Package from Report](#)<sup>[2260]</sup>

### 18.2.3 Web Style Templates

The *HTML and CSS Style Editor* enables you to edit the HTML associated with various sections of the HTML Report facility in Enterprise Architect. You would typically use this functionality to customize a report's look and feel for your company or client. The editor is derived from, and provides the facilities of, the common Code Editor.

**Access** [View](#) | [More Project Tools](#) | [Project Resources](#)

#### How to

To create or edit web style templates



| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See also |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | Open the Resources window and expand the <i>Templates</i> folder.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |          |
| 2    | <p>To:</p> <ul style="list-style-type: none"> <li>• Create a new template, right-click on the Web Style Templates folder and select the <b>Create HTML Template</b> context menu option; enter a name for the new template when prompted to do so</li> <li>• Edit an existing template, expand the Web Style Templates folder and either double-click on the template name or right-click and select the <b>Modify HTML Style Template</b> context menu option</li> </ul> <p>In either case, the HTML and CSS Style Editor displays.</p> <p>To delete a template, right-click on it and select the <b>Delete HTML Template</b> context menu option.</p> |          |

### Topics

| Topic        | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also                                                         |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
| <b>Usage</b> | <p>The HTML and CSS Style Editor contains a list of all available HTML fragments for modification and customization.</p> <p>Each fragment typically contains HTML plus one or more special tag names that Enterprise Architect replaces with information during generation. Currently you cannot alter the content within the tag names, but you can omit a complete tag by removing it, or alter its basic display properties in the surrounding HTML.</p> <p>Special tag names are delimited by # characters - for example, #NOTES#.</p> <p>The:</p> <ul style="list-style-type: none"> <li>• <b>Get Default</b> button retrieves the default Enterprise Architect template for the currently selected template item in the left hand list</li> <li>• <b>Save</b> button saves your version of the template for this style only</li> <li>• <b>Delete</b> button removes your modified version of the template, which causes Enterprise Architect to use the default template during report generation</li> </ul> <p>To select a template during generation, use the Style drop-down arrow on the Generate HTML Report dialog; once a style is selected, Enterprise Architect applies that to the current report.</p> <p>Select &lt;default&gt; for the inbuilt style.</p> | <a href="#">HTML Template Fragments</a><br><small>[2357]</small> |

### Notes

- Each time Enterprise Architect generates the web report it overwrites these files, so you must back up your modified versions and copy them back in after every update

Learn more

- [Editing Source Code](#)<sup>[1799]</sup>

Learning Center topics

- (Alt+F1) | **Reporting | HTML Templates**

**18.2.4 HTML Template Fragments**

This topic identifies each of the HTML Template fragments available through the **HTML and CSS Style Editor**, and defines the fields of each fragment.

| Section               | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Body - Diagram</b> | <ul style="list-style-type: none"> <li>• #AUTHOR# - Diagram author</li> <li>• #CREATEDATE# - Diagram created date</li> <li>• #CSS# - Style Sheet to use</li> <li>• #GUID# - Diagram GUID</li> <li>• #IMAGE# - Image of the diagram</li> <li>• #IMAGES# - Image path</li> <li>• #LOCKED# - Is the diagram locked (<b>true</b> or <b>false</b>)</li> <li>• #MODDATE# - Diagram modified date</li> <li>• #NAME# - Diagram name</li> <li>• #NOTE# - Diagram notes</li> <li>• #STEREOTYPE# - Diagram stereotype</li> <li>• #TITLE# - Diagram title</li> <li>• #TYPE# - Diagram type</li> <li>• #VERSION# - Diagram version</li> </ul>                                                     |
| <b>Body – Object</b>  | <ul style="list-style-type: none"> <li>• #ABSTRACT# - Element abstract (if true, = <b>abstract</b>)</li> <li>• #ACTIVE# - Element isActive (<b>true</b> or <b>false</b>)</li> <li>• #ALIAS# - Element alias ("<b>#ALIAS#</b>")</li> <li>• #AUTHOR# - Element author</li> <li>• #BEHAVIOR# - Returns the object behavior.</li> <li>• #CLASSIFIER# - Element classifier</li> <li>• #CLASSIFIERREF# - Returns the HREF of the classifier</li> <li>• #COMPLEXITY# - Element complexity</li> <li>• #CREATEDATE# - Element created date</li> <li>• #CSS# - Style Sheet to use</li> <li>• #DIAGRAMS# - List of diagrams the element is on</li> <li>• #DIFF# - Element difficulty</li> </ul> |

| Section                             | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                     | <ul style="list-style-type: none"> <li>• #GUID# - Element GUID</li> <li>• #IMAGES# - Image path</li> <li>• #KEYWORDS# - Element keywords</li> <li>• #LANGUAGE# - Element language</li> <li>• #LEAF# - Element isLeaf (<b>true</b> or <b>false</b>)</li> <li>• #LEVELNUMBER# - Element level number</li> <li>• #MODDATE# - Element modified date</li> <li>• #MULTIPLICITY# - Element multiplicity ("<b>Multiplicity: #MULTIPLICITY#</b>")</li> <li>• #NAME# - Element name</li> <li>• #NOTE# - Element notes</li> <li>• #PHASE# - Element phase</li> <li>• #PRIORITY# - Element priority</li> <li>• #ROOT# - Element isRoot (<b>true</b> or <b>false</b>)</li> <li>• #SCOPE# - Element scope</li> <li>• #STATUS# - Element status</li> <li>• #STEREOTYPE# - Element stereotype(s)</li> <li>• #TYPE# - Element type; for example, Class, Object</li> <li>• #VERSION# - Element version</li> </ul> |
| <b>Body – Project Glossary</b>      | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Body – Project Glossary Item</b> for each Project Glossary item</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Body - Project Glossary Item</b> | <ul style="list-style-type: none"> <li>• #MEANING# - Glossary Meaning</li> <li>• #TERM# - Glossary Term</li> <li>• #TYPE# - Glossary Type</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Body – Project Issue</b>         | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Body – Project Issue Item</b> for each Project Issue item</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Body - Project Issue Item</b>    | <ul style="list-style-type: none"> <li>• #DATERESOLVED# - Project Issue resolved date (blank if no date entered)</li> <li>• #ISSUE# - Project Issue name</li> <li>• #ISSUEDATE# - Project Issue issue date</li> <li>• #NOTES# - Project Issue notes</li> <li>• #OWNER# - Project Issue owner</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

| Section                          | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                  | <ul style="list-style-type: none"> <li>• #RESOLUTION# - Project Issue resolution</li> <li>• #RESOLVER# - Project Issue resolver</li> <li>• #STATUS# - Project Issue status</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Body – Project Task</b>       | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Body – Project Task Item</b> for each Project Task item</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Body - Project Task Item</b>  | <ul style="list-style-type: none"> <li>• #ENDDATE# - Project Task end date</li> <li>• #NAME# - Project Task name</li> <li>• #NOTES# - Project Task notes</li> <li>• #OWNER# - Project Task owner</li> <li>• #PHASE# - Project Task phase</li> <li>• #PRIORITY# - Project Task priority</li> <li>• #STARTDATE# - Project Task start date</li> <li>• #STATUS# - Project Task status</li> <li>• #TYPE# - Project Task type</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Content – Attributes</b>      | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Content – Attributes Item</b> for each attribute on an element</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Content – Attributes Item</b> | <ul style="list-style-type: none"> <li>• #ALIAS# - Attribute alias ("<i>&lt;i&gt;Alias:&lt;/i&gt; #ALIAS#&lt;br&gt;</i>")</li> <li>• #CONST# - Attribute is constant value ("<b>const</b> " &lt;-- Note Space)</li> <li>• #CONSTRAINT# - Attribute constraint</li> <li>• #DEFAULT# - Attribute default value ("<i>&lt;i&gt;Initial:&lt;/i&gt; #DEFAULT&lt;br&gt;</i>")</li> <li>• #NAME# - Attribute name</li> <li>• #NOTE# - Attribute notes</li> <li>• #ORDERED# - Attribute is ordered value ("<b>Ordered &lt;br /&gt;</b>")</li> <li>• #RANGE# - If lower != 1 ("<b>Range:&lt;lower&gt; to &lt;upper&gt;</b>")</li> <li>• #SCOPE# - Attribute scope ("<b>#SCOPE#</b> " &lt;-- Note space)</li> <li>• #STATIC# - Attribute is static value ("<b>static</b> " &lt;-- Note Space)</li> <li>• #STEREOTYPE# - Attribute stereotype(s)</li> <li>• #TAGS# - Attribute tags ("<b>Property Name=Property Value&lt;br&gt;</b>" )</li> <li>• #TYPE# - Attribute type (if <b>Column</b>, "<b>#TYPE(Column Precision, Scale)</b>")</li> </ul> |

| Section                                    | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Content – Constraints</b>               | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Content – Constraints Item</b> for each constraint on an element</li> </ul>                                                                                                                                                                                                                                                                                                                |
| <b>Content – Constraints Item</b>          | <ul style="list-style-type: none"> <li>• #CONSTRAINT# - Constraint name</li> <li>• #NOTES# - Constraint notes</li> <li>• #STATUS# - Constraint status</li> <li>• #TYPE# - Constraint type</li> </ul>                                                                                                                                                                                                                                                     |
| <b>Content – Custom Properties</b>         | <ul style="list-style-type: none"> <li>• #CONTENT# - List of all <b>Content – Custom Properties - Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                    |
| <b>Content – Custom Properties - Item</b>  | <ul style="list-style-type: none"> <li>• #PROPERTY# - Custom property name</li> <li>• #VALUE# - Custom property value<br/>           ( if type equal boolean<br/>             true or false<br/>             else<br/>             #VALUE#)</li> </ul>                                                                                                                                                                                                   |
| <b>Content – Files</b>                     | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Content - Files Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                      |
| <b>Content – Files Item</b>                | <ul style="list-style-type: none"> <li>• #DATE# - File date</li> <li>• #NAME# - File filename (&lt;a href="#"#FILENAME#" &lt;/a&gt;)</li> <li>• #NOTES - File notes</li> <li>• #SIZE# - File size</li> <li>• #TYPE# - File type</li> </ul>                                                                                                                                                                                                               |
| <b>Content - Inherited Attributes</b>      | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Content – Inherited Attributes Item</b> for each attribute on an element</li> </ul>                                                                                                                                                                                                                                                                                                        |
| <b>Content - Inherited Attributes Item</b> | <ul style="list-style-type: none"> <li>• #ALIAS# - Attribute alias ("<i>&lt;i&gt;Alias:&lt;/i&gt; #ALIAS#&lt;br&gt;</i>")</li> <li>• #CONST# - Attribute is constant value ("<b>const</b> " &lt;-- Note Space)</li> <li>• #CONSTRAINT# - Attribute constraint</li> <li>• #DEFAULT# - Attribute default value ("<i>&lt;i&gt;Initial:&lt;/i&gt; #DEFAULT&lt;br&gt;</i>")</li> <li>• #NAME# - Attribute name</li> <li>• #NOTE# - Attribute notes</li> </ul> |

| Section                                    | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                            | <ul style="list-style-type: none"> <li>• #ORDERED# - Attribute Is ordered value ("<b>Ordered</b> &lt;br /&gt;")</li> <li>• #RANGE# - If lower != 1 ("<b>Range:</b>&lt;lower&gt; to &lt;upper&gt;")</li> <li>• #SCOPE# - Attribute scope ("<b>#SCOPE#</b> " &lt;-- Note space)</li> <li>• #STATIC# - Attribute is static value ("<b>static</b> " &lt;-- Note Space)</li> <li>• #STEREOTYPE# - Attribute stereotype(s)</li> <li>• #TAGS# - Attribute tags ("<b>Property Name=Property Value</b>&lt;br&gt;")</li> <li>• #TYPE# - Attribute type (if <b>Column</b>, "<b>#TYPE(Column Precision, Scale)</b>")</li> </ul>                                                                                                                                                                                                                                                                                                              |
| <b>Content - Inherited Operations</b>      | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Content - Inherited Operations Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Content - Inherited Operations Item</b> | <ul style="list-style-type: none"> <li>• #ABSTRACT# - Operation abstract (<b>abstract</b>)</li> <li>• #ALIAS# - Operation alias ("<i>Alias</i>:&lt;/i&gt; <b>#ALIAS#</b>&lt;br&gt;")</li> <li>• #CONCURRENCY# - Operation concurrency (<b>blank</b> if not set)</li> <li>• #CONST# - Operation constant (<b>const</b>)</li> <li>• #CONSTRAINTS# - List of <b>Method Constraint</b></li> <li>• #ISQUERY# - Operation IsQuery (<b>isQuery</b>)</li> <li>• #NAME# - Operation name</li> <li>• #NOTE# - Operation notes.</li> <li>• #PARAMS# - List of <b>Content – Operations Item Parameters</b></li> <li>• #SCOPE# - Operation Scope</li> <li>• #STATIC# - Operation IsStatic (<b>static</b>)</li> <li>• #STEREOTYPE# - Operation stereotype</li> <li>• #TAGLABEL# - static text (<b>Tags</b>)</li> <li>• #TAGS# - Attribute tags ("<b>Property Name=Property Value</b>&lt;br&gt;")</li> <li>• #TYPE# - Operation type</li> </ul> |
| <b>Content – Notes</b>                     | <ul style="list-style-type: none"> <li>• #VALUE#&gt; - Notes text</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Content – Operations</b>                | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Content - Operations Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Content – Operations Item</b>           | <ul style="list-style-type: none"> <li>• #ABSTRACT# - Operation abstract (<b>abstract</b>)</li> <li>• #ALIAS# - Operation alias ("<i>Alias</i>:&lt;/i&gt; <b>#ALIAS#</b>&lt;br&gt;")</li> <li>• #CONCURRENCY# - Operation concurrency (<b>blank</b> if not set)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

| Section                                     | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                             | <ul style="list-style-type: none"> <li>• #CONST# - Operation constant (<b>const</b>)</li> <li>• #CONSTRAINTS# - List of <b>Method Constraint</b></li> <li>• #ISQUERY# - Operation IsQuery (<b>isQuery</b>)</li> <li>• #NAME# - Operation name</li> <li>• #NOTE# - Operation notes.</li> <li>• #PARAMS# - List of <b>Content – Operations Item Parameters</b></li> <li>• #SCOPE# - Operation Scope</li> <li>• #STATIC# - Operation IsStatic (<b>static</b>)</li> <li>• #STEREOTYPE# - Operation stereotype</li> <li>• #TAGLABEL# - static text (<b>Tags</b>)</li> <li>• #TAGS# - Attribute tags ("<b>Property Name=Property Value&lt;br&gt;</b>")</li> <li>• #TYPE# - Operation type</li> </ul> |
| <b>Content – Operations Item Parameters</b> | <ul style="list-style-type: none"> <li>• #DEFAULT# - Op Parameter default</li> <li>• #GUID# - Op Parameter GUID</li> <li>• #KIND# - Op Parameter kind</li> <li>• #NAME# - Op Parameter name</li> <li>• #NOTES# - Op Parameter notes</li> <li>• #STEREOTYPE# - Op Parameter stereotype</li> <li>• #TYPE# - Op Parameter type</li> </ul>                                                                                                                                                                                                                                                                                                                                                         |
| <b>Content – Resource Allocation</b>        | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Content – Resource Allocation Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Content – Resource Allocation Item</b>   | <ul style="list-style-type: none"> <li>• #ACTUAL# - Resource actual time</li> <li>• #ENDDATE# - Resource end date</li> <li>• #EXPECTED# - Resource expected date</li> <li>• #NOTES# - Resource notes</li> <li>• #PERCENT# - Resource percent complete</li> <li>• #RESOURCE# - Resource name</li> <li>• #ROLE# - Resource role</li> <li>• #STARTDATE# - Resource start date</li> <li>• #TIME# - Resource time</li> </ul>                                                                                                                                                                                                                                                                        |
| <b>CSS – Main</b>                           | <ul style="list-style-type: none"> <li>• None</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

| Section                        | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CSS – ToC</b>               | <ul style="list-style-type: none"> <li>• None</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Feature Notes</b>           | <ul style="list-style-type: none"> <li>• #FIELD#</li> <li>• #VALUE#</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Javascript</b>              | <ul style="list-style-type: none"> <li>• None</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Link (Association)</b>      | <ul style="list-style-type: none"> <li>• #CONTENT#</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Link (Association) Item</b> | <ul style="list-style-type: none"> <li>• #CONNECTION# - Connector type</li> <li>• #IMAGE# - The file path of the images</li> <li>• #LINK# - (&lt;a href= "path to element"&gt; "Connection Name" &lt;/a&gt;)</li> <li>• #NOTES# - The connector notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #SOURCEROLE# - Source role</li> <li>• #SOURCEROLENOTES# - Source role notes</li> <li>• #STEREOTYPE# - Connector stereotype</li> <li>• #TARGETROLE# - Target role</li> <li>• #TARGETROLENOTES# Target role notes</li> <li>• #TYPE# - Connector type</li> </ul>           |
| <b>Link (Flow)</b>             | <ul style="list-style-type: none"> <li>• #CONTENT#</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Link (Flow) Item</b>        | <ul style="list-style-type: none"> <li>• #DIRECTION# - Connector direction</li> <li>• #ELEMNAME# - Name of the element at the source/destination of the connector</li> <li>• #IMAGE# - The file path of the images</li> <li>• #LINK# - (&lt;a href= "path to element"&gt; "Connection Name" &lt;/a&gt;)</li> <li>• #LINKREF# - Page name of the element at the source/destination of the connector</li> <li>• #NAME# - Connector name</li> <li>• #NOTES# - The connector notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #STEREOTYPE# - Connector stereotype</li> </ul> |



| Section                        | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                | <ul style="list-style-type: none"> <li>• #TYPE# - Connector type</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Link (Other)</b>            | <ul style="list-style-type: none"> <li>• #CONTENT#</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Link (Other) Item</b>       | <ul style="list-style-type: none"> <li>• #CONNECTION# - Connector type</li> <li>• #IMAGE# - The file path of the images</li> <li>• #LINK# - (&lt;a href= "path to element"&gt; "Connection Name" &lt;/a&gt;)</li> <li>• #NOTES# - The connector notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #SOURCEROLE# - Source role</li> <li>• #SOURCEROLENOTES# - Source role notes</li> <li>• #STEREOTYPE# - Connector stereotype</li> <li>• #TARGETROLE# - Target role</li> <li>• #TARGETROLENOTES# Target role notes</li> <li>• #TYPE# - Connector type</li> </ul> |
| <b>Linked Document</b>         | <ul style="list-style-type: none"> <li>• #LINKDOC# - Linked Document.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Linked Requirement</b>      | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Linked Requirement Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Linked Requirement Item</b> | <ul style="list-style-type: none"> <li>• #DIFF# - Linked Requirement difficulty</li> <li>• #NAME# - Linked Requirement name</li> <li>• #PRIORITY# - Linked Requirement priority</li> <li>• #STATUS# - Linked Requirement status</li> </ul>                                                                                                                                                                                                                                                                                                                                                         |
| <b>Linked Section</b>          | <ul style="list-style-type: none"> <li>• #ITEMS#</li> <li>• #TITLE#</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Maintenance</b>             | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Maintenance Line Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Maintenance Line Item</b>   | <ul style="list-style-type: none"> <li>• #DATEREPORTED# - Maintenance date reported</li> <li>• #DATERESOLVED# - Maintenance date resolved</li> <li>• #IMAGE# - The file path of the images</li> <li>• #NOTES# - Maintenance notes</li> </ul>                                                                                                                                                                                                                                                                                                                                                       |

| Section                        | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                | <ul style="list-style-type: none"> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #PRIORITY# - Maintenance priority</li> <li>• #PROBLEM# - Maintenance name</li> <li>• #REPORTEDBY# - Maintenance reported by</li> <li>• #RESOLVEDBY# - Maintenance resolved by</li> <li>• #RESOLVERNOTES# - Maintenance resolved notes</li> <li>• #STATUS# - Maintenance status</li> <li>• #TYPE# - Maintenance type</li> <li>• #VERSION# - Maintenance version</li> </ul> |
| <b>Message</b>                 | <p>(Applies only to Sequence messages.)</p> <ul style="list-style-type: none"> <li>• #CONTENT# - Loops the <b>Message Item</b> for each attribute on an element</li> <li>• #DIRECTION# - Contains the value <b>To</b> or <b>From</b></li> </ul>                                                                                                                                                                                                                                        |
| <b>Message Item</b>            | <ul style="list-style-type: none"> <li>• #KIND# - The Message <b>Kind</b> field</li> <li>• #MESSAGE# - Connector Message</li> <li>• #NAME# - Name of the Message (&lt;a href="&lt;path&gt;"&gt;#NAME#)<br/>If Message has a classifier: #NAME#="#NAME# :Classifier"</li> <li>• #NOTES# - The Message notes (&lt;strong&gt;Type:&lt;/strong&gt; #Item Type#&lt;br /&gt; #NOTES#)</li> <li>• #SYNCH# - The Message <b>Synch</b> field</li> <li>• #TYPE# - The type of Message</li> </ul> |
| <b>Method Constraint</b>       | <ul style="list-style-type: none"> <li>• #NAME# - Method Constraint name</li> <li>• #NOTES# - Method Constraint notes</li> <li>• #TYPE# - Method Constraint type</li> </ul>                                                                                                                                                                                                                                                                                                            |
| <b>Object Requirement</b>      | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Object Requirement Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Object Requirement Item</b> | <ul style="list-style-type: none"> <li>• #DIFF# - Requirement difficulty</li> <li>• #NAME# - Requirement name</li> <li>• #NOTES# - Requirement notes</li> <li>• #PRIORITY# - Requirement priority</li> <li>• #STABILITY# - Requirement stability</li> <li>• #STATUS# - Requirement status</li> </ul>                                                                                                                                                                                   |

| Section                        | Fields                                                                                                                                                                                                                                                                    |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                | <ul style="list-style-type: none"> <li>• #TYPE# - Requirement type</li> </ul>                                                                                                                                                                                             |
| <b>Package Content</b>         | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Package Content Row</b></li> </ul>                                                                                                                                                                        |
| <b>Package Content Row</b>     | <ul style="list-style-type: none"> <li>• #NAME# - Link to Package (&lt;a href="#Link to file#"&gt;#Package name#&lt;/a&gt;)</li> <li>• #TYPE# - Link to Image (&lt;img src="#path to image#"&gt;)</li> </ul>                                                              |
| <b>Page - Basic template</b>   | <ul style="list-style-type: none"> <li>• #CONTENT# - Contains <b>Body - Diagram</b> through to <b>Body - Object</b></li> <li>• #TITLE# - Current package name</li> </ul>                                                                                                  |
| <b>Page - Index</b>            | <ul style="list-style-type: none"> <li>• #CSS# - Style Sheet to use</li> <li>• #HOME# - A link to the Start page</li> <li>• #JS# - Javascript to use</li> <li>• #TITLE# - Current package name</li> <li>• #TOC# - <b>To be established</b></li> </ul>                     |
| <b>Page - ToC</b>              | <ul style="list-style-type: none"> <li>• None</li> </ul>                                                                                                                                                                                                                  |
| <b>Scenario</b>                | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Scenario Item</b></li> <li>• #EXCEPTIONS# - List of Structured Scenario exceptions</li> <li>• #STRUCTURED# - List of Structured Scenarios</li> </ul>                                                      |
| <b>Scenario Item</b>           | <ul style="list-style-type: none"> <li>• #IMAGE# - The file path of the images</li> <li>• #NOTES# - Scenario notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #SCENARIO# - Scenario name</li> <li>• #TYPE# - Scenario type</li> </ul> |
| <b>Scenario Exception</b>      | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Scenario Exception Item</b> for each exception</li> </ul>                                                                                                                                                   |
| <b>Scenario Exception Item</b> | <ul style="list-style-type: none"> <li>• #NAME# - Exception name</li> <li>• #STEPNO# - Exception step number</li> </ul>                                                                                                                                                   |

| Section                          | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                  | <ul style="list-style-type: none"> <li>• #TYPE# - Exception Type</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Scenario Structured</b>       | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Scenario Structured Items</b> for each Structured Scenario item</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Scenario Structured Items</b> | <ul style="list-style-type: none"> <li>• #ACTION# - Name of the scenario</li> <li>• #STEPNO# - Scenario step number</li> <li>• #RESULT# - Step result value</li> <li>• #USES# - Step uses value</li> <li>• #STATE# - Step state value</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Tagged Value</b>              | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Tagged Value Line Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Tagged Value Line Item</b>    | <ul style="list-style-type: none"> <li>• #IMAGE# - The file path of the images</li> <li>• #NOTES# - Tagged Value notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #PROPERTY# - Tagged Value name</li> <li>• #VALUE# - Tagged Value if type is boolean (value is <b>true</b> or <b>false</b>)</li> </ul>                                                                                                                                                                                                                                                                                                                                               |
| <b>Test Cases</b>                | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Test Cases Line Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Test Cases Line Item</b>      | <ul style="list-style-type: none"> <li>• #ACCEPTANCE# - Test case acceptance notes</li> <li>• #CHECKEDBY# - Test case checked by</li> <li>• #CLASS# - Test case class (<b>Unit, Integration, System, Acceptance, Scenario</b>)</li> <li>• #IMAGE# - The file path of the images</li> <li>• #INPUT# - Test case input notes</li> <li>• #NOTES# - Test case notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #RESULTS# - Test case result notes</li> <li>• #RUNBY# - Test case run by</li> <li>• #RUNDATE# - Test case last run</li> <li>• #STATUS# - Test case status</li> <li>• #TEST# - Test case name</li> <li>• #TYPE# - Test case type</li> </ul> |

| Section | Fields |
|---------|--------|
|         |        |

#### Learn more

- [HTML and CSS Style Editor](#)<sup>[2355]</sup>

### 18.2.5 Master Documents and Model Documents

Enterprise Architect enables you to generate HTML documentation using Master Document and Model Document elements, just as you do for **RTF reports**. However, there are two differences:

- You do not specify the template in the document elements; you must select it on the Generate HTML Report dialog, in the **Style** field
- You do not use the *SearchName* and *SearchValue* Tagged Values in the Model Document elements to select the generated elements

#### Learn more

- [Virtual Documents](#)<sup>[2327]</sup>
- [Generate HTML Report Dialog](#)<sup>[2354]</sup>

#### Learning Center topics

- **(Alt+F1) | Reporting | Virtual Documents**

#### 18.2.5.1 Generate a Master Document (HTML)

To generate the HTML documentation defined in the Master Document and/or Model Documents, follow the steps below

#### How to

| Step | Action                                                                                                                                                                                                             | See also                                                      |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| 1    | On the documentation diagram, click on the Master Document element (or on an independent Model Document element).                                                                                                  |                                                               |
| 2    | Select the <b>Project   Documentation   HTML Report</b> option.<br>The Generate HTML Report dialog displays.                                                                                                       |                                                               |
| 3    | Set the options for your HTML document as required. Ensure that you select your Master Document template in the <b>Style</b> field.                                                                                | <a href="#">Generate HTML Report Dialog</a> <sup>[2354]</sup> |
| 4    | Click on the <b>Generate</b> button to create the documentation.<br>The HTML Report Generator works through the defined content of the Master Document element and/or the Model Document elements and pulls in the |                                                               |

| Step     | Action                                                                                                          | See also |
|----------|-----------------------------------------------------------------------------------------------------------------|----------|
|          | information from the listed packages, formatted according to the template identified in the <b>Style</b> field. |          |
| <b>5</b> | Click on the <b>View</b> button to view the documentation.                                                      |          |

#### Learning Center topics

- (Alt+F1) | [Reporting | Virtual Documents](#)

**Part**



## 19 Automation



This section describes how you can automate and extend the facilities of Enterprise Architect through:

| Topic                                 | Link                                                                    |
|---------------------------------------|-------------------------------------------------------------------------|
| Scripts                               | <a href="#">Scripts</a> <sup>[2372]</sup>                               |
| The Enterprise Architect Object Model | <a href="#">The Enterprise Architect Object Model</a> <sup>[2381]</sup> |
| The Enterprise Architect Add-In Model | <a href="#">The Enterprise Architect Add-In Model</a> <sup>[2585]</sup> |

### Learning Center topics

- (Alt+F1) | **Automation** | **Automation** | **Introduction to Automation**
- (Alt+F1) | **Automation** | **Addins**
- (Alt+F1) | **Automation** | **Scripting**



## 19.1 Scripting



### Topics

| Topic        | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See also                                                                                                                                                                                                                          |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Usage</b> | <p>Scripts executed in Enterprise Architect have access to the currently open model and are a powerful tool for querying and updating the model in situations that would otherwise require you to perform time consuming and repetitive GUI tasks.</p> <p>Enterprise Architect supports management of scripts using the following script engines:</p> <ul style="list-style-type: none"> <li>• JavaScript</li> <li>• Microsoft JScript</li> <li>• Microsoft VBScript</li> </ul> <p>The management interface for Scripting is the Scripting window, which contains the:</p> <ul style="list-style-type: none"> <li>• Script Tree View (Scripts tab), which you use to review, create and edit scripts</li> <li>• Script Console (Console tab), which you use to operate on an executing script.</li> </ul> <p>Scripts are managed in groups: the first group in the list is always Local Scripts, which are files in the Scripts subdirectory of the Enterprise Architect installation - any instance of Enterprise Architect that has a currently open model can see these scripts; you cannot create, edit, drag-and-drop or delete Local scripts.</p> <p>All other groups are User Scripts, which you create yourself; a user group can be <b>one of several types</b>, each of which applies a template and certain conditions to the scripts you create within that group.</p> <p>User scripts are only visible inside the model in which they were created; the contents of the scripts are stored with the model, although they can be saved to the file system easily using the Script Editor.</p> | <p><a href="#">Scripts Tab</a> <sup>[2373]</sup></p> <p><a href="#">Console Tab</a> <sup>[2376]</sup></p> <p><a href="#">Script Group Properties</a> <sup>[2375]</sup></p> <p><a href="#">Script Editor</a> <sup>[2378]</sup></p> |

### Notes

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering

and Ultimate editions

- If you intend to use the Scripting facility under Crossover/WINE, you must also install Internet Explorer version 6.0 or above

#### Learning Center topics

- (Alt+F1) | **Automation | Scripting**

### 19.1.1 Scripts Tab













#### Topics




| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See also                                                                                                                        |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| <b>Usage</b>         | <p>The Scripts tab is composed of a toolbar and a view of all scripts according to group.</p> <p>To execute a script, press ( <b>Ctrl</b> ) while you double-click on the script name.</p> <p>To write or edit a user script, double-click on the script name to display the Script Editor; this usually displays a script template, determined by the user group type as assigned on creation or on the Script Group Properties dialog.</p> <p>You can move or copy a script from one user scripts folder to another; to:</p> <ul style="list-style-type: none"> <li>• Move a script, highlight it in the Scripts tab and drag it into the user scripts folder it now belongs to</li> <li>• Copy a script, highlight it in the Scripts tab and press ( <b>Ctrl</b> ) while you drag it into the user scripts folder in which to duplicate it</li> </ul> | <p><a href="#">Script Group Properties</a> <small>[2375]</small></p> <p><a href="#">Script Editor</a> <small>[2378]</small></p> |
| <b>Context Menus</b> | <p>The script groups and their scripts also have context menus that provide some or all of the following options:</p> <ul style="list-style-type: none"> <li>• <b>Group Properties</b> - to display or edit script group properties in the Script Group Properties dialog</li> <li>• <b>Run Script</b> - to execute the selected script</li> <li>• <b>Edit Script</b> - to update the selected script</li> <li>• <b>Rename Script</b> - to change the name of the selected group or script</li> <li>• <b>New VBScript/JScript/JavaScript</b> - add a new script to the selected user group</li> <li>• <b>Import Workflow Script</b> - to display the Browser dialog through which you locate and select a workflow script source (.vbs) file to import into the Workflow script folder</li> </ul>                                                        | <p><a href="#">Script Group Properties</a> <small>[2375]</small></p>                                                            |

| Topic | Detail                                                                                                                       | See also |
|-------|------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <ul style="list-style-type: none"> <li>• <b>Delete Group/Script</b> - to delete the selected user group or script</li> </ul> |          |

The Scripts tab toolbar provides the following options:

### Reference

| Icon                                                                                | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See also |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|    | <p>Create a new script group; this option displays a short menu of the types of script group you can create, namely:</p> <ul style="list-style-type: none"> <li>• Normal Group ()</li> <li>• Project Browser Group ()</li> <li>• Diagram Group ()</li> <li>• Workflow Group ()</li> <li>• Search Group ()</li> <li>• Model Search Group</li> </ul> <p>The new group is added to the end of the list in the Scripting window, with the 'New group' text highlighted so that you can type in the group name.</p> |          |
|  | <p>Create a new script file in the selected script group; displays a short menu of the types of script you can create, namely:</p> <ul style="list-style-type: none"> <li>• VBScript ()</li> <li>• JScript ()</li> <li>• JavaScript ()</li> </ul> <p>The new script is added to the end of the list in the selected group, with the 'New script' text highlighted so that you can type in the script name.</p>                                                                                                                                                                                                                                                                |          |
|  | <p>Refresh the script tree in the Scripting window; this icon also reloads any changes made to a workflow script.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |          |
|  | <p>Compile and execute the selected script.</p> <p>The output from the script is written to the Script tab of the Output window, which you display using the <b>View Script Output</b> button (below).</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |          |



| Icon                                                                              | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See also                                |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
|  | Stop an executing script; the icon is disabled if no script is executing.                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                         |
|  | <p>Delete a <i>script</i> from the model; you cannot use this icon to delete a <i>script group</i> (see the <i>Context Menu</i> item above), scripts in the Local Scripts group, or a script that is executing.</p> <p>The system prompts you to confirm the deletion only if the <b>Confirm Deletes</b> checkbox is selected in the Project Browser panel of the General page of the Options dialog; if this option is not selected, no prompt is displayed.</p> <p>Script deletion is permanent - scripts cannot be recovered.</p> | <a href="#">General setting</a><br> 544 |
|  | Display the Output window with the results of the most recently executed script displayed in the Script tab.                                                                                                                                                                                                                                                                                                                                                                                                                         |                                         |





### Notes

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions
- If you add, delete or change a script, you might have to reload the model in order for the changes to take effect
- If you select to delete a script group that contains scripts, the system always prompts you to confirm the action regardless of any system settings for delete operations; be certain that you intend to delete the group and its scripts before confirming the deletion - deletion of script groups and scripts is permanent

### 19.1.1.1 Script Group Properties

#### Topics

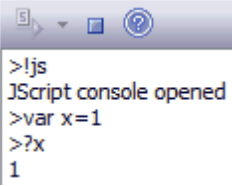
| Topic        | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See also                                                                                                                                   |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Usage</b> | <p>The Script Group Properties dialog enables you to set the following properties of the script group:</p> <ul style="list-style-type: none"> <li>• <b>Name</b> - The name of the script group; the dialog also displays the automatically-assigned GUID of the group</li> <li>• <b>Group Type</b> - The type of scripts contained in the group; this can be one of: <ul style="list-style-type: none"> <li>• <b>Normal</b> - () Ordinary scripts</li> <li>• <b>Project Browser</b> - () Scripts that are listed in and can be executed from the Project Browser <b>Scripts</b> context menu option</li> </ul> </li> </ul> | <a href="#">Package Context Menu</a><br> 571 <br><br><a href="#">Workflow Scripts</a><br> 298 <br><br><a href="#">Model Search</a><br> 611 |



| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See also                                              |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
|       | <ul style="list-style-type: none"> <li>• <b>Workflow</b> - (  ) Scripts executed by Enterprise Architect's workflow engine; you can create only VB scripts of this type</li> <li>• <b>Search</b> - (  ) Scripts that can be executed as <b>model searches</b>; these scripts are listed in the <b>Search</b> field of the Model Search window, in the last category in the list</li> <li>• <b>Diagram</b> - (  ) Scripts that are listed in and can be executed from the Scripts submenu of the <b>Diagram</b> context menu</li> <li>• <b>Model Search</b> - (  ) Scripts that can be executed from a search context menu, on the results of a successfully-executed SQL search that includes CLASSGUID and CLASSTYPE, or a Query-built search</li> <li>• <b>Notes</b> - Your own notes on the script group</li> </ul> | <a href="#">Diagram Context Menu</a> <sup>[687]</sup> |

## 19.1.2 Console Tab

### Topics

| Topic                   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See also |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Abstract</b>         | <p>The script console is a tab of the Scripting window; it is a command line interpreter through which you can quickly enable a script engine and enter commands to act on the script.</p> <p>You type the commands in the field at the bottom of the tab; when you press the ( <b>Enter</b> ) key, the script console executes the commands and displays any output immediately</p> <p>You can input two types of command:</p> <ul style="list-style-type: none"> <li>• Console commands</li> <li>• Script commands</li> </ul> |          |
| <b>Console Commands</b> | <p>Console commands are preceded by the ! character and instruct the console to perform an action.</p> <p>The available console commands are listed below; to list these commands on the <b>Console</b> tab itself, type ? in the console field (without the preceding ! character) and press the ( <b>Enter</b> ) key.</p> <ul style="list-style-type: none"> <li>• <b>c(lear)</b> - clears the console display</li> </ul>                                                                                                     |          |

| Topic                      | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See also |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                            | <ul style="list-style-type: none"> <li>• <b>sa(ve)</b> - saves the console display to a file</li> <li>• <b>h(elp)</b> - prints a list of commands, as for ?</li> <li>• <b>VB</b> - opens a VBScript console</li> <li>• <b>JA</b> - opens a JavaScript console</li> <li>• <b>JS</b> - opens a JScript console</li> <li>• <b>st(op)</b> - closes any script running console</li> <li>• <b>i(nclude) name</b> - executes the named script item; name is of the format <i>GroupName.ScriptName</i> (spaces are allowed in names)</li> <li>• <b>?</b> - (without the !) lists commands</li> <li>• <b>?name</b> - Outputs the value of a variable name (only if a script console is opened).</li> </ul>                                                                                                                                                                                                                                |          |
| <b>Script Commands</b>     | <p>A script command is script code that depends on the script engine.</p> <p>Script commands can be executed only once a script console has been created.</p> <p><i>Examples:</i></p> <p>The following lines, entered into the console, create a <i>VBScript</i> console and then execute the script <i>MyScript</i> in the user group <i>MyGroup</i>:</p> <pre>&gt;! VB &gt;! i MyGr oup. My Scr i pt</pre> <p>The following lines, entered into the console, create a <i>JScript</i> console and then create a variable called <i>x</i> with the value <b>1</b>:</p> <pre>&gt;! JS &gt;var x = 1</pre> <p>The following image shows the result of entering the above <i>JScript</i> example; remember that you can use <b>?&lt;variable name&gt;</b> to get the current value of any item you have created during the console session.</p>  |          |
| <b>Console Tab Toolbar</b> | The <b>Console</b> tab has two operations available through the toolbar:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |          |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See also |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <ul style="list-style-type: none"> <li>• <b>Open Console</b> (  ) - click on the down-arrow and select to open a VBScript console, JScript console or JavaScript console</li> <li>• <b>Stop Script</b> (  ) - click to stop an executing script and close the current console</li> </ul> |          |

### Notes

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions
- You can save the output of the console to an external .txt file; right-click on the console window, select the **Save As** context menu option, browse for an appropriate file location and specify the file name

### 19.1.3 Script Editor

The *Script Editor* enables you to edit scripts, and to run and stop an open script.

The editor is based on, and provides the facilities of, the common Code Editor. The editor opens in the main work area.

**Access** [View | Scripting > Scripts: double-click on script name](#)

### Use to

- Save changes to the current script
- Save the current script under a different name
- Run the script
- Stop the executing script
- View the script output in the Scripts tab of the Output window

### Topics

| Topic                                      | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See also                                                     |
|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| <b>Enterprise Architect Script Objects</b> | <p>Enterprise Architect adds to the available functionality and features of the editor script language by providing inbuilt objects; these are either <i>Type Libraries</i>, providing intellisense for editing purposes, or <i>Runtime objects</i>, providing access to objects of the types described in the Type Libraries.</p> <p>The available intellisense scripting objects are:</p> <ul style="list-style-type: none"> <li>• EA</li> <li>• MathLib</li> </ul> | <a href="#">Automation Interface</a><br><small>[238]</small> |

| Topic                                                        | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | See also |
|--------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                              | <ul style="list-style-type: none"> <li>• System</li> <li>• The runtime scripting objects (below)</li> </ul> <p>The available runtime scripting objects are:</p> <ul style="list-style-type: none"> <li>• Repository ( Type: IDualRepository ) - this is the Enterprise Architect <b>Automation Interface</b></li> <li>• Maths ( Type: IMath )</li> <li>• Session ( Type: ISession )</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |          |
| <b>Script Editing<br/>Intellisense<br/>(Required Syntax)</b> | <p>Intellisense is available not only in the Script Editor, but also in the Script Console; Intellisense at its most basic is presented for the inbuilt functionality of the script engine.</p> <p>For intellisense on the additional Enterprise Architect scripting objects listed above, you must declare variables according to syntax that specifies a type; it is not necessary to use this syntax to execute a script properly, it is only present so that the correct intellisense can be displayed for an item.</p> <p>The syntax can be seen in, for example:</p> <pre>Di m e as EA. El ement</pre> <p>Then when you type, in this case, <b>e.</b>, the editor displays a list of member functions and properties of <b>e</b>'s type.</p> <p>You select one of these to complete the line of script; you might, therefore, type:</p> <pre>VBTr ace( e.</pre> <p>As you type the period, the editor presents the appropriate list and you might double-click on, for example, <b>Abstract</b>; this is inserted in the line, and you continue to type or select the rest of the statement, in this case adding the end space and parenthesis:</p> <pre>VBTr ace( e. Abst ract )</pre> |          |
| <b>Keystrokes</b>                                            | <p>In the Script Editor or Console, intellisense is presented on the following keystrokes.</p> <ul style="list-style-type: none"> <li>• Press ( . ) (period) after an item to list any members for that item's type</li> <li>• Press ( <b>Ctrl</b> ) + ( <b>Spacebar</b> ) on a word to list any intellisense items with a name starting with the string at the point the keys were pressed</li> <li>• Press ( <b>Ctrl</b> ) +( <b>Spacebar</b> ) when not on a word to display any available top level intellisense items -</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |          |



| Topic | Detail                                                                                                                         | See also |
|-------|--------------------------------------------------------------------------------------------------------------------------------|----------|
|       | these are the intellisense objects described above plus any built-in methods and properties of the current scripting language. |          |

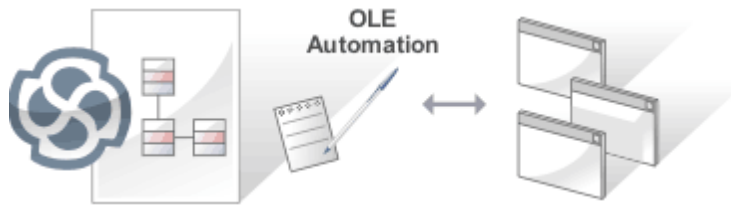
### Notes

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions
- Enterprise Architect's scripting supports declaring variables to match the Enterprise Architect types; this enables the editor to present **intellisense**, but is not necessary for executing the script

### Learn more

- [Editing Source Code](#)<sup>[1799]</sup>
- [Scripts Tab](#)<sup>[2379]</sup>

## 19.2 Enterprise Architect Object Model



### Topics

| Topic                                                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | See also                                                                                                                                                          |
|------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Introduction</b>                                  | <p>Automation provides a way for other applications to access the information in an Enterprise Architect model using Windows OLE Automation (ActiveX). Typically this involves scripting clients such as MS Word or Visual Basic, or using scripts created within Enterprise Architect using the Scripting window.</p> <p>The Automation Interface provides a way of accessing the internals of Enterprise Architect models. Examples of things you can do using the Automation Interface include:</p> <ul style="list-style-type: none"> <li>• Perform repetitive tasks, such as update the version number for all elements in a model</li> <li>• Generate code from a State Machine diagram</li> <li>• Produce custom reports</li> <li>• Perform ad hoc queries</li> </ul> | <a href="#">Scripting</a> <sup>[2372]</sup>                                                                                                                       |
| <b>Connecting to the Automation Interface</b>        | All development environments capable of generating <i>ActiveX</i> Com clients should be able to connect to the Enterprise Architect Automation Interface. This guide provides detailed instructions on connecting to the interface using Microsoft Visual Basic 6.0, Borland Delphi 7.0, Microsoft C# and Java. There are also more detailed steps on how to set-up Visual Basic; the principles are applicable to other languages.                                                                                                                                                                                                                                                                                                                                          | <a href="#">Connecting to the Interface</a> <sup>[2382]</sup><br><a href="#">Set References in Visual Basic</a> <sup>[2385]</sup>                                 |
| <b>Examples and Tips</b>                             | Instruction on how to use the Automation Interface is provided by means of sample code. See pointers to the samples and other available resources. Also, consult the extensive Reference Section.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <a href="#">Pointers to the Samples</a> <sup>[2386]</sup><br><a href="#">Available Resources</a> <sup>[2389]</sup><br><a href="#">Reference</a> <sup>[2389]</sup> |
| <b>Calling Executables from Enterprise Architect</b> | Enterprise Architect can be set up to call an external application. You can pass parameters on the current position selected in the Project Browser to the application being called. For instructions, go to the Call from Enterprise Architect topic. A more sophisticated method is to create Add-Ins, which are discussed in a separate topic.                                                                                                                                                                                                                                                                                                                                                                                                                            | <a href="#">Call from Enterprise Architect</a> <sup>[2387]</sup><br><a href="#">Add-Ins</a> <sup>[2585]</sup>                                                     |

## 19.2.1 Using the Automation Interface

This section provides instructions on how to connect to and use the Automation Interface, including:

| Topic                              | Link                                                                 |
|------------------------------------|----------------------------------------------------------------------|
| Connecting to the Interface        | <a href="#">Connecting to the Interface</a> <sup>[2382]</sup>        |
| Setting References In Visual Basic | <a href="#">Setting References In Visual Basic</a> <sup>[2385]</sup> |
| Examples and Tips                  | <a href="#">Examples and Tips</a> <sup>[2386]</sup>                  |

### 19.2.1.1 Connect to the Interface

All development environments capable of generating ActiveX Com clients should be able to connect to the Enterprise Architect Automation Interface.

By way of example, the following sections describe how to connect using several such tools. The procedure might vary slightly with different versions of these products.

#### How To - Microsoft Visual Basic 6.0:

| Step | Action                                                                                                                                                                                                                                                             | See also |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | Create a new project.                                                                                                                                                                                                                                              |          |
| 2    | Select the <b>Project   References</b> menu option.                                                                                                                                                                                                                |          |
| 3    | Select <i>Enterprise Architect Object Model 2.0</i> from the list.<br>If this does not appear, go to the command line and re-register Enterprise Architect using:<br><br><i>EA.exe /unregister</i><br><br>then<br><br><i>EA.exe /register</i>                      |          |
| 4    | See the general library documentation on the use of Classes. The following example creates and opens a repository object:<br><br><pre>Public Sub ShowRepository()<br/>    Dim MyRep As New EA.Repository<br/>    MyRep.OpenFile "c:\eat est.eap"<br/>End Sub</pre> |          |

**How To - Borland Delphi 7.0:**

| Step | Action                                                                                                                                                                                                                                                                                                          | See also |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | Create a new project.                                                                                                                                                                                                                                                                                           |          |
| 2    | Select the <b>Project   Import Type Library</b> menu option.                                                                                                                                                                                                                                                    |          |
| 3    | Select <b>Enterprise Architect Object Model 2.0</b> from the list.<br>If this does not appear, go to the command line and re-register Enterprise Architect using:<br><br><i>EA.exe /unregister</i><br><br>then<br><br><i>EA.exe /register</i>                                                                   |          |
| 4    | Click on the <b>Create Unit</b> button.                                                                                                                                                                                                                                                                         |          |
| 5    | Include <i>EA_TLB</i> in Project1's <i>Uses</i> clause.                                                                                                                                                                                                                                                         |          |
| 6    | See the general library documentation on the use of Classes. The following example creates and opens a repository object:<br><br><pre>procedure TForm1.Button1Click(Sender: TObject); var   r: TRepository;   b: boolean; begin   r := TRepository.Create(nil);   b := r.OpenFile('c:\eat\eat.eap'); end;</pre> |          |

**How To - Microsoft C#:**

| Step | Action                                                                                                             | See also |
|------|--------------------------------------------------------------------------------------------------------------------|----------|
| 1    | Select the Visual Studio <b>Project   Add Reference</b> menu option.                                               |          |
| 2    | Click on the Browse tab.                                                                                           |          |
| 3    | Navigate to the folder in which you installed Enterprise Architect; usually:<br><br>Program Files/Sparx Systems/EA |          |

| Step | Action                                                                                                                                                                                                                                                                                             | See also |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|      | <p>Select</p> <p>Interop.EA.dll</p>                                                                                                                                                                                                                                                                |          |
| 4    | <p>See the general library documentation on the use of Classes. The following example creates and opens a repository object:</p> <pre>private void button1_Click(object sender, System. Event Args e) {     EA.Repository r = new EA.RepositoryClass();     r.OpenFile("c:\\eat est.eap"); }</pre> |          |

**How To - Java:**

| Step | Action                                                                                                                                                                                                                                                                                                                                                          | See also                                                                 |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| 1    | <p>Copy the file:</p> <p>SSJavaCOM.dll</p> <p>from the Java API subdirectory of your installed directory, usually:</p> <p>Program Files/Sparx Systems/EA</p> <p>into any location within the Windows PATH</p> <p>windows\system32 directory.</p>                                                                                                                |                                                                          |
| 2    | <p>Copy the file</p> <p>eaapi.jar</p> <p>from the Java API subdirectory of your installed directory, usually:</p> <p>Program Files/Sparx Systems/EA</p> <p>to a location in the Java CLASSPATH or where the Java class loader can find it at run time.</p>                                                                                                      |                                                                          |
| 3    | <p>All of the Classes described in the documentation are in the package <i>org.sparx</i>. See the <b>general library documentation</b> for their use. The following example creates and opens a repository object:</p> <pre>public void OpenRepository() {     org.sparx.Repository r = new org.sparx.Repository ();     r.OpenFile("c:\\eat est.eap"); }</pre> | <a href="#">Reference</a><br><span style="font-size: small;">2389</span> |

| Step | Action | See also |
|------|--------|----------|
|      |        |          |

### 19.2.1.1.1 Set References In Visual Basic

This topic describes how to use the Enterprise Architect ActiveX interface with Visual Basic (VB). Use is ensured for Visual Basic version 6, but might vary slightly with versions other than version 6.

It is assumed that you have accessed VB through a Microsoft Application such as VB 6.0, MS Word or MS Access. If the code is not called from within Word, the *Word VB* reference must also be set.

#### How to

On creating a new VB project, to set a reference to an Enterprise Architect Type Library and a Word Type Library follow the steps below:

| Step | Action                                                                                                 | See also |
|------|--------------------------------------------------------------------------------------------------------|----------|
| 1    | Select the <b>Tools   References</b> menu option.                                                      |          |
| 2    | Select the <b>Enterprise Architect Object Model 2.10</b> checkbox from the list.                       |          |
| 3    | Do the same for VB or VB Word: select the checkbox for the <b>Microsoft Word 10.0 Object Library</b> . |          |
| 4    | Click on the <b>OK</b> button.                                                                         |          |

#### Notes

- If Enterprise Architect Object Model 2.10 does not appear in the list, go to the command line and manually re-enter Enterprise Architect using the following:
  - To unregister Enterprise Architect: `ea.exe /unregister`
  - To register Enterprise Architect: `ea.exe /register`
- **Visual Basic 5/6** users should also note that the version number of the Enterprise Architect interface is stored in the VBP project file in a form similar to the following:

```
Reference=* \ G{ 64FB2BF4- 9EFA- 11D2- 8307- C45586000000} #2. 2#0#.. \ . \ . \ . \ . \ . \ .
\ Program Files\ Sparx Systems\ EA\ EA. TLB#Enterprise Architect Object
Model 2. 02
```

If you experience problems moving from one version of Enterprise Architect to another, open the VBP file in a text editor and remove this line, then open the project in Visual Basic and use Project-

References to create a new reference to the Enterprise Architect Object model

Reference to objects in Enterprise Architect and Word should now be available in the **Object Browser**, which can be accessed from the main menu by selecting **View | Object Browser**, or by pressing ( **F2** )

The drop-down list on the top-left of the window should now include Enterprise Architect and Word; if MS-Project is installed this must also be set up

### 19.2.1.2 Examples and Tips

#### Topics

| Topic                  | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See also                                                                                                                                                                                                                                                                                                                               |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Usage</b>           | <p>Instructions for using the interface are provided through sample code. There are several sets of examples:</p> <ul style="list-style-type: none"> <li>• VB 6 and C# examples are available in the Code Samples folder under your Enterprise Architect installation<br/>(default: C: \ P r o g r a m F i l e s \ S p a r x S y s t e m s \ E A \ C o d e S a m p l e s )</li> <li>• Enterprise Architect can be set up to <b>call an external application</b></li> <li>• Several VB.NET code snippets are provided in the <b>reference section</b></li> <li>• A comprehensive example of using Visual Basic to create MS Word documentation is available from the internet at <a href="http://www.sparxsystems.com/resources/developers/autint_vb.html">www.sparxsystems.com/resources/developers/autint_vb.html</a></li> <li>• Additional samples are available from the Sparx Systems website; see the <i>Available Resources</i> topic</li> </ul> | <p><a href="#">Call an External Application</a> <sup>[2387]</sup></p> <p><a href="#">Code Samples</a> <sup>[2569]</sup></p> <p><a href="http://www.sparxsystems.com/resources/developers/autint_vb.html">www.sparxsystems.com/resources/developers/autint_vb.html</a></p> <p><a href="#">Available Resources</a> <sup>[2389]</sup></p> |
| <b>Tips and Tricks</b> | <p>Additionally, you should note the following tips and tricks:</p> <ul style="list-style-type: none"> <li>• An instance of the Enterprise Architect (<i>EA.exe</i>) process is executed when you initialize a new repository object - this process must remain running in order to perform automation tasks; if the main window is visible, you can safely minimize it, but it must remain running</li> <li>• The Enterprise Architect ActiveX Interface is a functional interface rather than a data interface; when you load data through the interface there is a</li> </ul>                                                                                                                                                                                                                                                                                                                                                                       | <p><a href="#">Repository Class</a> <sup>[2448]</sup></p>                                                                                                                                                                                                                                                                              |

| Topic                                          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also                                                  |
|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
|                                                | <p>noticeable delay as Enterprise Architect user interface elements (such as Windows and menus) are loaded and the specified database connection is established</p> <ul style="list-style-type: none"> <li>• Collections use a zero-based index; for example, <code>Repository.Models(0)</code> represents the first model in the repository</li> <li>• During the development of your client software your program might terminate unexpectedly and leave <i>EA.exe</i> running in such a state that it is unable to support further interface calls; if your program terminates abnormally, ensure that Enterprise Architect is not left running in the background (see the Windows Task Manager / Process tab)</li> <li>• A handle to a currently running instance of Enterprise Architect can be obtained through the use of a <code>GetObject()</code> call (see the reference page for the <i>App</i> object); accessing your Enterprise Architect model via the <i>App</i> object enables querying the current User Interface status, such as using <code>GetCurrentItem()</code> on the <b>Repository</b> object to detect the current selection by the user, allowing for rapid prototyping and testing</li> </ul> | <p><a href="#">App Class</a> <sup>[2412]</sup></p>        |
| <p><b>Enterprise Architect Not Closing</b></p> | <p>If your automation controller was written using the .NET framework, Enterprise Architect does not close even after you release all your references to it.</p> <p>To force the release of the COM pointers, call the memory management functions as shown below:</p> <pre>GC. Collect (); GC. Wait For Pending Finalizers ();</pre> <p>There are additional concerns when controlling a running instance of Enterprise Architect that loads Add-Ins - see the <i>Tricks and Traps</i> topic for details.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <p><a href="#">Tricks and Traps</a> <sup>[2591]</sup></p> |

### 19.2.1.2.1 Call from Enterprise Architect

Enterprise Architect can be set up to call an external application. You can pass parameters on the current position selected in the Project Browser to the application being called.

Access **Tools | Customize > Tools**



**Topics**

| Topic       | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See also |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Uses</b> | <p>With this you can:</p> <ul style="list-style-type: none"> <li>• Add a command line for an application</li> <li>• Define parameters to pass to this application</li> </ul> <p>The parameters required for running the AutInt executable are:</p> <ul style="list-style-type: none"> <li>• The Enterprise Architect file parameter \$f and</li> <li>• The current PackageID \$p</li> </ul> <p>Hence the arguments should simply contain: \$f , \$p</p> |          |

The available parameters for passing information to external applications are:

| Parameter | Description                                | Notes                                              |
|-----------|--------------------------------------------|----------------------------------------------------|
| \$d       | Diagram ID                                 | ID for accessing associated diagram.               |
| \$D       | Diagram GUID                               | GUID for accessing the associated diagram.         |
| \$e       | Comma separated list of element IDs        | All elements selected in the current diagram.      |
| \$E       | Comma separated list of element GUIDs      | All elements selected in the current diagram.      |
| \$f       | Project Name                               | For example: C: \ p r o j e c t s \ EAexample.eap. |
| \$F       | Calling Application (Enterprise Architect) | 'Enterprise Architect'.                            |
| \$p       | Current Package ID                         | For example: 144.                                  |
| \$P       | Package GUID                               | GUID for accessing this package.                   |

Once this has been set up, the application can be called from the main menu in Enterprise Architect using the **Extensions | <YourApplication>** menu option.

### 19.2.1.2.2 Available Resources

Other available resources include:

| Resource                                                                                                                      | Download Link                                                                                                                                                  |
|-------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VB 6 Add-In for generating MS Word documentation.                                                                             | <a href="http://www.sparxsystems.com/resources/developers/autint_vb.html">www.sparxsystems.com/resources/developers/autint_vb.html</a>                         |
| VB 6 Add-In to display a custom ActiveX graph control within the Enterprise Architect window as a new view.                   | <a href="http://www.sparxsystems.com/resources/developers/autint_vb_custom_view.html">www.sparxsystems.com/resources/developers/autint_vb_custom_view.html</a> |
| A basic Add-In framework written in C#. Useful as a starting point for authoring your own custom Enterprise Architect Add-In. | <a href="http://www.sparxsystems.com/bin/CS_AddinFramework.zip">www.sparxsystems.com/bin/CS_AddinFramework.zip</a>                                             |
| An extension on the <i>CS_AddinFramework</i> example showing how to export Tagged Values to a .csv file.                      | <a href="http://www.sparxsystems.com/bin/CS_AddinTaggedCSV.zip">www.sparxsystems.com/bin/CS_AddinTaggedCSV.zip</a>                                             |
| A basic Add-In skeleton written in Delphi.                                                                                    | <a href="http://www.sparxsystems.com/bin/DelphiDemo.zip">www.sparxsystems.com/bin/DelphiDemo.zip</a>                                                           |
| A simple example Add-In written in C#.                                                                                        | <a href="http://www.sparxsystems.com/bin/CS_Sample.zip">www.sparxsystems.com/bin/CS_Sample.zip</a>                                                             |










#### Learn more

- For further information, see [www.sparxsystems.com/resources/developers/autint.html](http://www.sparxsystems.com/resources/developers/autint.html)

## 19.2.2 Reference






This section provides detailed information on all the objects available in the object model provided by the Automation Interface, covering:

| Topic                      | Link                                                         |
|----------------------------|--------------------------------------------------------------|
| Interface Overview Package | <a href="#">Interface Overview Package</a> <sup>[2390]</sup> |
| App Object                 | <a href="#">App Object</a> <sup>[2412]</sup>                 |
| Enumerations               | <a href="#">Enumerations</a> <sup>[2413]</sup>               |

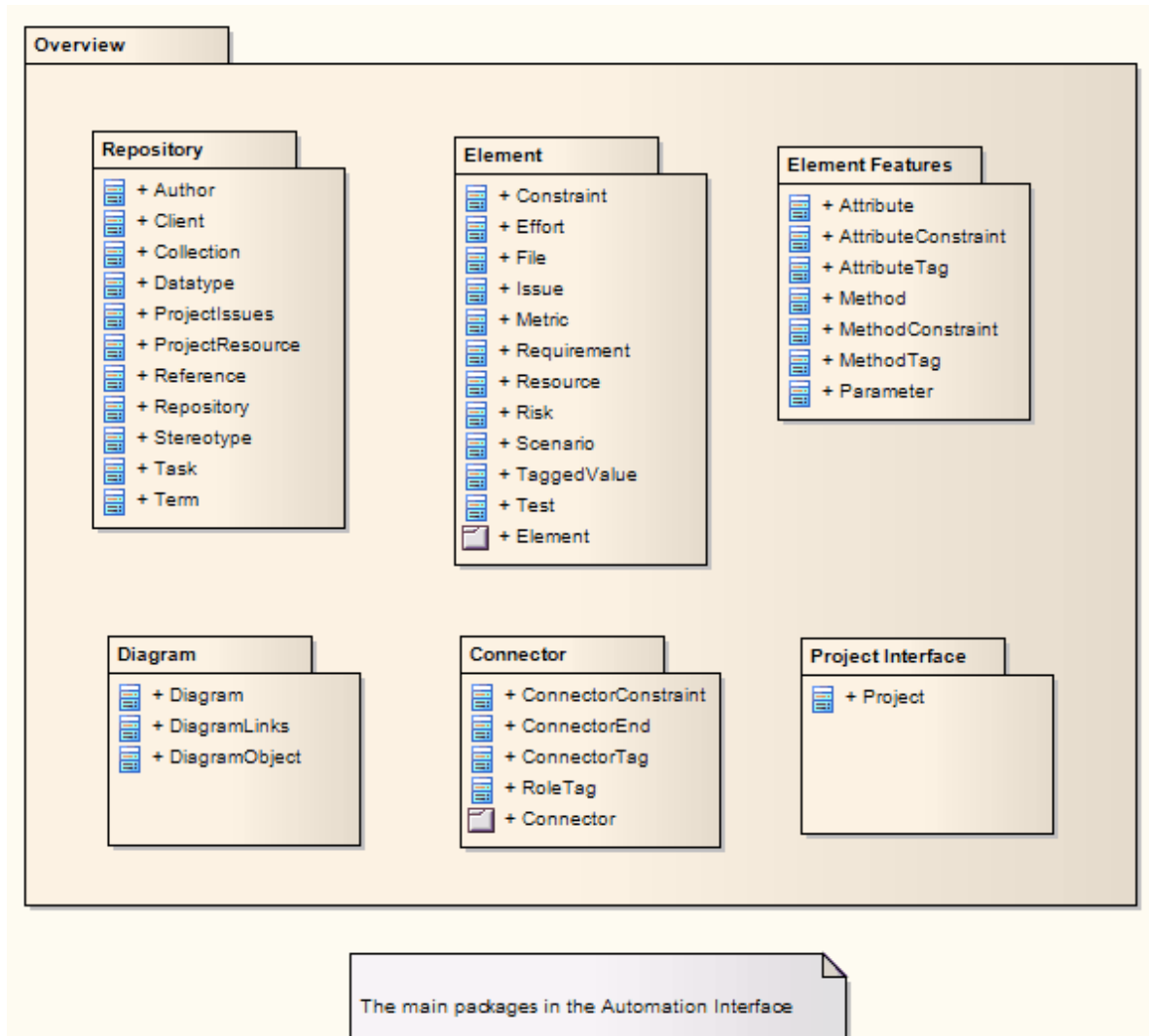
| Topic                                | Link                                                                                                                                     |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Repository Package                   | <a href="#">Repository Package</a>                    |
| Element Package                      | <a href="#">Element Package</a>                       |
| Element Features Package             | <a href="#">Element Features Package</a>              |
| Connector Package                    | <a href="#">Connector Package</a>                     |
| Diagram Package                      | <a href="#">Diagram Package</a>                       |
| Project Interface Package            | <a href="#">Project Interface Package</a>             |
| Document Generator Interface Package | <a href="#">Document Generator Interface Package</a>  |
| Mail Interface Package               | <a href="#">Mail Interface Package</a>              |
| Code Samples                         | <a href="#">Code Samples</a>                        |

### 19.2.2.1 Interface Overview Package

This package provides an overview of the main components of the Automation Interface. These include the:

| Topic                           | Detail                                                                                | Link                                                                                                                           |
|---------------------------------|---------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository Package</b>       | Represents the model as a whole and provides entry to model packages and collections. | <a href="#">Repository Package</a>        |
| <b>Element Package</b>          | Identifies the basic structural units (such as Class, Use Case and Object).           | <a href="#">Element Package</a>           |
| <b>Element Features Package</b> | Identifies the attributes and operations defined on an element.                       | <a href="#">Element Features Package</a>  |
| <b>Diagram Package</b>          | Describes the visible drawings contained in the model.                                | <a href="#">Diagram Package</a>           |
| <b>Connector Package</b>        | Defines the relationships between elements.                                           | <a href="#">Connector Package</a>         |

The following diagram illustrates the main interface packages and their associated contents. Each element in this document is creatable by Automation and can be accessed through the various collections that exist or, in some cases, directly.



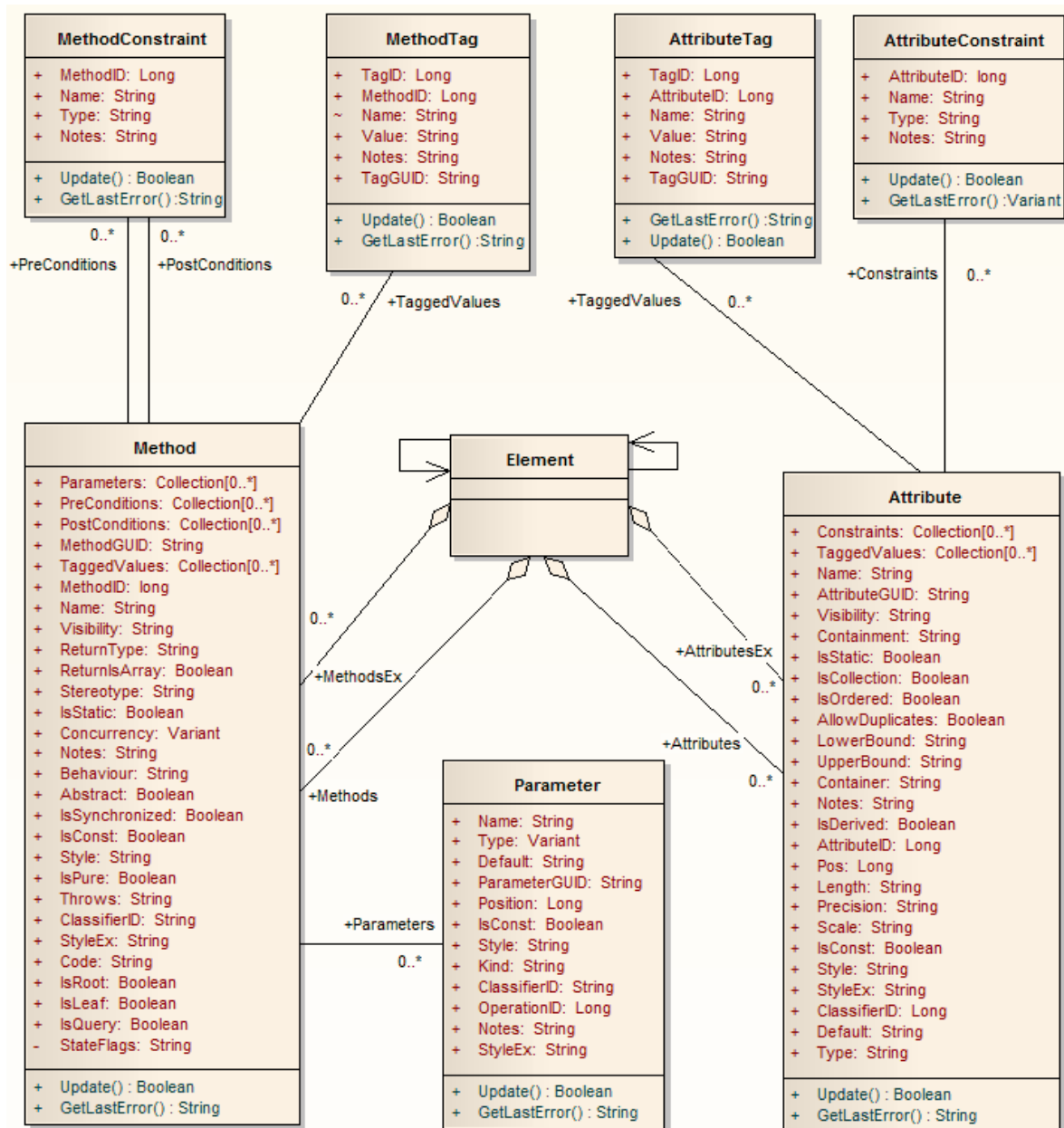
The *Repository* Class is the starting point for all use of the automation interface. It contains the high level system objects and entry point into the model itself using the *Models* collection and the other system level collections.

### 19.2.2.1.1 Element Features Package

The *ElementFeatures* package contains descriptions of the model interfaces that enable access to operations and attributes, and their associated Tagged Values and constraints.

This diagram illustrates the components associated with element features. These include *Attributes* and *Methods*, and the associated constraints and Tagged Values related to them. It also includes the *Parameter* object that defines the arguments associated with an operation (method).

#### Example



### 19.2.2.1.1.1 Attribute Class

An *attribute* corresponds to a UML Attribute. It contains further collections for constraints and Tagged Values. Attributes are accessed from the Element *Attributes* collection.

**Associated table in .EAP file:** *t\_attribute*

#### Attribute Attributes:

| Attribute              | Type           | Notes                                                                |
|------------------------|----------------|----------------------------------------------------------------------|
| <b>AllowDuplicates</b> | <i>Boolean</i> | Read/Write<br>Indicates if duplicates are allowed in the collection. |

| Attribute            | Type                                               | Notes                                                                                                                                                          |
|----------------------|----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      |                                                    | If the attribute represents a database column this, when set, represents the <b>Not Null</b> option.                                                           |
| <b>AttributeGUID</b> | <i>String</i>                                      | Read only<br>A globally unique ID for the current attribute. This attribute is system generated.                                                               |
| <b>AttributeID</b>   | <i>Long</i>                                        | Read only<br>The local ID number of the attribute.                                                                                                             |
| <b>ClassifierID</b>  | <i>Long</i>                                        | Read/Write<br>The classifier ID, if appropriate, indicating the base type associated with the attribute, if not a primitive type.                              |
| <b>Container</b>     | <i>String</i>                                      | Read/Write<br>The container type.                                                                                                                              |
| <b>Containment</b>   | <i>String</i>                                      | Read/Write<br>The type of containment - <b>Not Specified</b> , <b>By Reference</b> or <b>By Value</b> .                                                        |
| <b>Constraints</b>   | <a href="#">Collection</a><br><small>[242]</small> | Read only<br>A collection of <i>AttributeConstraint</i> objects, used to access and manage constraints associated with this attribute.                         |
| <b>Default</b>       | <i>String</i>                                      | Read/Write<br>The initial value assigned to this attribute.                                                                                                    |
| <b>IsCollection</b>  | <i>Boolean</i>                                     | Read/Write<br>Indicates if the current feature is a collection or not. If the attribute represents a database column this, when set, represents a Foreign Key. |
| <b>IsConst</b>       | <i>Boolean</i>                                     | Read/Write<br>A flag indicating if the attribute is <b>Const</b> or not.                                                                                       |
| <b>IsDerived</b>     | <i>Boolean</i>                                     | Read/Write<br>Indicates if the attribute is derived (that is, a calculated value).                                                                             |

| Attribute         | Type                                                | Notes                                                                                                                                                                           |
|-------------------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>IsOrdered</b>  | <i>Boolean</i>                                      | Read/Write<br>Indicates if a collection is ordered or not. If the attribute represents a database column this, when set, represents a Primary Key.                              |
| <b>IsStatic</b>   | <i>Boolean</i>                                      | Read/Write<br>Indicates if the current attribute is a static feature or not. If the attribute represents a database column this, when set, represents the <b>Unique</b> option. |
| <b>Length</b>     | <i>String</i>                                       | Read/Write<br>The attribute length, where applicable.                                                                                                                           |
| <b>LowerBound</b> | <i>String</i>                                       | Read/Write<br>A value for the collection lower boundary.                                                                                                                        |
| <b>Name</b>       | <i>String</i>                                       | Read/Write<br>The attribute name.                                                                                                                                               |
| <b>Notes</b>      | <i>String</i>                                       | Read/Write<br>Further notes on this attribute.                                                                                                                                  |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                                                     |
| <b>ParentID</b>   | <i>Long</i>                                         | Read only<br>Returns the <i>ElementID</i> of the element that this attribute is a part of.                                                                                      |
| <b>Pos</b>        | <i>Long</i>                                         | Read/Write<br>The position of the attribute in the Class attribute list.                                                                                                        |
| <b>Precision</b>  | <i>String</i>                                       | Read/Write<br>The precision value.                                                                                                                                              |
| <b>Scale</b>      | <i>String</i>                                       | Read/Write<br>The scale value.                                                                                                                                                  |
| <b>Stereotype</b> | <i>String</i>                                       | Read/Write                                                                                                                                                                      |

| Attribute             | Type                                                                                   | Notes                                                                                                                                                |
|-----------------------|----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       |                                                                                        | Sets or gets the stereotype for this attribute.                                                                                                      |
| <b>StereotypeEx</b>   | <i>String</i>                                                                          | Read/Write<br>Provides all the applied stereotypes of the attribute, in a comma-separated list.                                                      |
| <b>Style</b>          | <i>String</i>                                                                          | Read/Write<br>Contains the <b>Alias</b> property for this attribute.                                                                                 |
| <b>StyleEx</b>        | <i>String</i>                                                                          | Read/Write<br>Advanced style settings, reserved for the use of Sparx Systems.                                                                        |
| <b>TaggedValues</b>   | <a href="#">Collection</a><br>[2427] of type<br><a href="#">AttributeTag</a><br>[2397] | Read only<br>A collection of <i>AttributeTag</i> objects, used to access and manage Tagged Values associated with this attribute.                    |
| <b>TaggedValuesEx</b> | <a href="#">Collection</a><br>[2427] of type<br><a href="#">TaggedValue</a><br>[2504]  | Read only<br>A collection of <i>TaggedValue</i> objects belonging to the current attribute and the <i>TaggedValuesEx</i> property of its classifier. |
| <b>Type</b>           | <i>String</i>                                                                          | Read/Write<br>The attribute type (by name; also see <i>ClassifierID</i> ).                                                                           |
| <b>UpperBound</b>     | <i>String</i>                                                                          | Read/Write<br>A value for the collection upper boundary.                                                                                             |
| <b>Visibility</b>     | <i>String</i>                                                                          | Read/Write<br>Identifies the scope of the attribute - <b>Private</b> , <b>Protected</b> , <b>Public</b> or <b>Package</b> .                          |

**Attribute Methods:**

| Method                 | Type          | Notes                                                                                                                                                                                 |
|------------------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i> | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |



| Method           | Type           | Notes                                                                                                                                                                          |
|------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Update ()</b> | <i>Boolean</i> | Updates the current attribute object after modifying or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

### Notes

- The Attribute Class in Enterprise Architect does not have an **Alias** property; you can access the **Alias** property of the Attribute Class using the **Style** property

#### 19.2.2.1.1.2 *AttributeConstraint* Class

An *AttributeConstraint* is a constraint associated with the current Attribute.

Associated table in .EAP file: *t\_attributeconstraints*

#### AttributeConstraint Attributes:

| Attribute          | Type                                         | Notes                                                                       |
|--------------------|----------------------------------------------|-----------------------------------------------------------------------------|
| <b>AttributeID</b> | <i>Long</i>                                  | Read/Write<br>The ID of the attribute this constraint applies to.           |
| <b>Name</b>        | <i>String</i>                                | Read/Write<br>The name of the constraint.                                   |
| <b>Notes</b>       | <i>String</i>                                | Read/Write<br>Descriptive notes about the constraint.                       |
| <b>ObjectType</b>  | <a href="#">ObjectType</a> <sup>[2419]</sup> | Read only<br>Distinguishes objects referenced through a Dispatch interface. |
| <b>Type</b>        | <i>String</i>                                | Read/Write<br>The type of constraint.                                       |

#### AttributeConstraint Methods:

| Method                 | Type           | Notes                                                                                                                                                                                             |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used, as an exception is thrown when an error occurs.            |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current <i>AttributeConstraint</i> object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

### 19.2.2.1.1.3 AttributeTag Class

An *AttributeTag* represents a Tagged Value associated with an attribute.

**Associated table in .EAP file:** *t\_attributetag*

#### AttributeTag Attributes:

| Attribute          | Type                                                | Notes                                                                                                                                                                                       |
|--------------------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AttributeID</b> | <i>Long</i>                                         | Read/Write<br>The local ID of the attribute associated with this Tagged Value.                                                                                                              |
| <b>FQName</b>      | <i>String</i>                                       | Read only<br>The fully-qualified name of the tag.                                                                                                                                           |
| <b>Name</b>        | <i>String</i>                                       | Read/Write<br>The name of the tag.                                                                                                                                                          |
| <b>Notes</b>       | <i>String</i>                                       | Read/Write<br>Further descriptive notes about this tag.<br><br>If <b>Value</b> (below) is set to " <b>&lt;memo&gt;</b> ", then <b>Notes</b> should contain the actual Tagged Value content. |
| <b>ObjectType</b>  | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                                                                 |
| <b>TagGUID</b>     | <i>String</i>                                       | Read/Write<br>A globally unique ID for this Tagged Value.                                                                                                                                   |

| Attribute    | Type          | Notes                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>TagID</b> | <i>Long</i>   | Read only<br>The local ID to identify the Tagged Value.                                                                                                                                                                                                                                                                                                                                          |
| <b>Value</b> | <i>String</i> | Read/Write<br>The value assigned to this tag.<br>This field has a 255 character limit. If the value is greater than 255 characters long, set the value to "<memo>" and insert the body of text in the <b>Notes</b> attribute (above).<br>When reading existing Tagged Values, if <b>Value</b> = "<memo>" then the developer should read the actual body of text from the <b>Notes</b> attribute. |

**AttributeTag Methods:**

| Method                                                  | Type           | Notes                                                                                                                                                                                   |
|---------------------------------------------------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetAttribute (string propName)</b>                   | <i>String</i>  | Returns the text of a single named property within a structured tagged value.                                                                                                           |
| <b>GetLastError()</b>                                   | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br>This function is rarely used as an exception is thrown when an error occurs.       |
| <b>HasAttributes()</b>                                  | <i>Boolean</i> | Returns true if the tagged values is a structured tagged value with one or more properties.                                                                                             |
| <b>SetAttribute (string propName, string propValue)</b> | <i>Boolean</i> | Sets the text of a single named property within a structured tagged value.                                                                                                              |
| <b>Update ()</b>                                        | <i>Boolean</i> | Updates the current <i>AttributeTag</i> object after modification or appending a new item.<br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

#### 19.2.2.1.1.4 CustomProperties Collection

The *CustomProperties* collection contains 0 or more *CustomProperties* associated with the current element. These properties provide advanced UML configuration options, and must not be added to or deleted. The value of each property can be set.

##### CustomProperty:

| Attribute         | Type                                                | Notes                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>       | <i>String</i>                                       | Read only<br>The CustomProperty name.                                                                                                                                                                                                                                                                                                                                 |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                                                                                                                                                           |
| <b>Value</b>      | <i>String</i>                                       | Read/Write<br>The value associated with this custom property. This can be: <ul style="list-style-type: none"> <li>• A string</li> <li>• The boolean values <b>true</b> or <b>false</b>, or</li> <li>• An enumeration value from a defined list</li> </ul> <p>The UML 2.3 specification in general provides information on the kinds of enumeration relevant here.</p> |

##### Notes

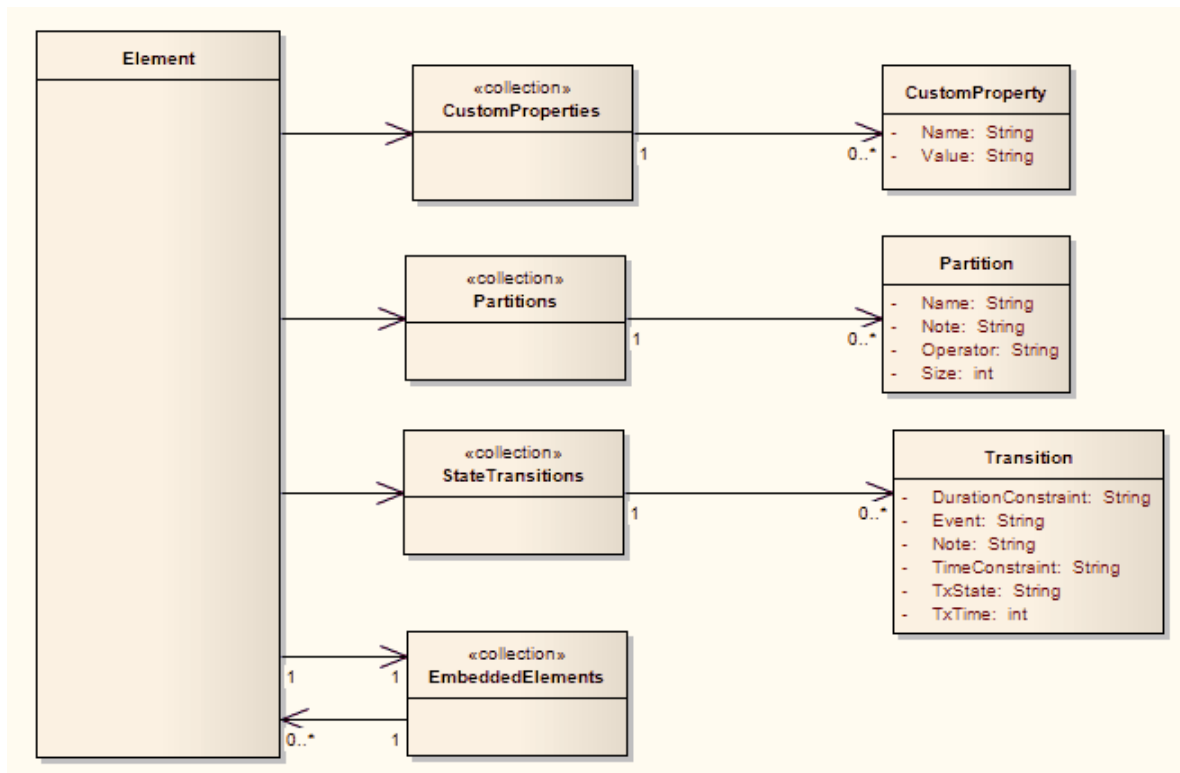
- The number and type of properties vary depending on the actual element

#### 19.2.2.1.1.5 EmbeddedElements Collection

In UML 2.3 an element can have one or more embedded elements such as Ports, Pins, Parameters or ObjectNodes. These are attached to the boundary of the host element and cannot be moved off the element. They are owned by their host element. This collection gives easy access to the set of elements embedded on the surface of an element. Note that some embedded elements can have their own embedded element collection (for example, Ports can have Interfaces embedded on them).

The *EmbeddedElements* collection contains Element objects.

##### Example



### 19.2.2.1.1.6 Method Class

A *method* represents a UML *operation*. It is accessed from the Element *Methods* collection and includes collections for parameters, constraints and Tagged Values.

**Associated table in .EAP file:** *t\_operation*

#### Method Attributes:

| Attribute           | Type           | Notes                                                                                                                                                                                                                                                                                                       |
|---------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Abstract</b>     | <i>Boolean</i> | Read/Write<br>A flag indicating if the method is abstract (1) or not (0).                                                                                                                                                                                                                                   |
| <b>Behavior</b>     | <i>String</i>  | Read/Write<br>Some further explanatory behavior notes (for example, pseudocode).<br>In earlier releases of Enterprise Architect this attribute had the UK/ Australian spelling 'Behaviour'; this is still present for backwards compatibility, but please now use the 'Behavior' attribute for consistency. |
| <b>ClassifierID</b> | <i>String</i>  | Read/Write<br>The Classifier ID that applies to the <i>ReturnType</i> .                                                                                                                                                                                                                                     |

| Attribute             | Type           | Notes                                                                                                |
|-----------------------|----------------|------------------------------------------------------------------------------------------------------|
| <b>Code</b>           | <i>String</i>  | Read/Write<br>An optional field to hold the method code (used for the <b>Initial Code</b> field).    |
| <b>Concurrency</b>    | <i>Variant</i> | Read/Write<br>Indicates the concurrency type of the method.                                          |
| <b>IsConst</b>        | <i>Boolean</i> | Read/Write<br>A flag indicating that the method is <b>Const</b> .                                    |
| <b>IsLeaf</b>         | <i>Boolean</i> | Read/Write<br>A flag to indicate if the method is <i>Leaf</i> (cannot be overridden).                |
| <b>IsPure</b>         | <i>Boolean</i> | Read/Write<br>A flag indicating that the method is defined as <b>Pure</b> in C++.                    |
| <b>IsQuery</b>        | <i>Boolean</i> | Read/Write<br>A flag to indicate if the method is a query (that is, does not alter Class variables). |
| <b>IsRoot</b>         | <i>Boolean</i> | Read/Write<br>A flag to indicate if the method is <i>Root</i> .                                      |
| <b>IsStatic</b>       | <i>Boolean</i> | Read/Write<br>A flag to indicate a static method.                                                    |
| <b>IsSynchronized</b> | <i>Boolean</i> | Read/Write<br>A flag indicating a Synchronized method call.                                          |
| <b>MethodGUID</b>     | <i>String</i>  | Read/Write<br>A globally unique ID for the current method. This is system generated.                 |
| <b>MethodID</b>       | <i>Long</i>    | Read only<br>A local ID for the current method, only valid within this .EAP file.                    |
| <b>Name</b>           | <i>String</i>  | Read/Write<br>The method name.                                                                       |

| Attribute             | Type                                 | Notes                                                                                                                                           |
|-----------------------|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
|                       |                                      |                                                                                                                                                 |
| <b>Notes</b>          | <i>String</i>                        | Read/Write<br>Descriptive notes on the method.                                                                                                  |
| <b>ObjectType</b>     | <a href="#">ObjectType</a><br>[2415] | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                     |
| <b>Parameters</b>     | <a href="#">Collection</a><br>[2427] | Read only<br>The <i>Parameters</i> collection for the current method, used to add and access parameter objects for the current method.          |
| <b>ParentID</b>       | <i>Long</i>                          | Read only<br>Returns the <i>ElementID</i> of the element that this method belongs to.                                                           |
| <b>Pos</b>            | <i>Long</i>                          | Read/Write<br>Specifies the position of the method within the set of operations defined for a Class.                                            |
| <b>PostConditions</b> | <a href="#">Collection</a><br>[2427] | Read only<br>The PostConditions (constraints) as they apply to this method. This returns a <i>MethodConstraint</i> object of type <b>post</b> . |
| <b>PreConditions</b>  | <a href="#">Collection</a><br>[2427] | Read only<br>The PreConditions (constraints) as they apply to this method. This returns a <i>MethodConstraint</i> object of type <b>pre</b> .   |
| <b>ReturnsArray</b>   | <i>Boolean</i>                       | Read/Write<br>A flag to indicate that the return value is an array.                                                                             |
| <b>ReturnType</b>     | <i>String</i>                        | Read/Write<br>The return type for the method; this can be a primitive data type or a Class or Interface type.                                   |
| <b>StateFlags</b>     | <i>String</i>                        | Read/Write<br>Some flags as applied to methods in State elements.                                                                               |
| <b>Stereotype</b>     | <i>String</i>                        | Read/Write                                                                                                                                      |

| Attribute           | Type                                                                                                      | Notes                                                                                                                     |
|---------------------|-----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
|                     |                                                                                                           | The method stereotype (optional).                                                                                         |
| <b>StereotypeEx</b> | <i>String</i>                                                                                             | Read/Write<br>All the applied stereotypes of the method in a comma-separated list.                                        |
| <b>Style</b>        | <i>String</i>                                                                                             | Read/Write<br>Contains the <b>Alias</b> property for this method.                                                         |
| <b>StyleEx</b>      | <i>String</i>                                                                                             | Read/Write<br>Advanced style settings, reserved for the use of Sparx Systems.                                             |
| <b>TaggedValues</b> | <a href="#">Collection</a> <sup>[2427]</sup><br>of type<br><a href="#">MethodTag</a><br><sup>[2405]</sup> | Read only<br>The <i>TaggedValues</i> collection for the current method. This accesses a list of <i>MethodTag</i> objects. |
| <b>Throws</b>       | <i>String</i>                                                                                             | Read/Write<br>Exception information.                                                                                      |
| <b>Visibility</b>   | <i>String</i>                                                                                             | Read/Write<br>The method scope - <b>Public</b> , <b>Protected</b> , <b>Private</b> or <b>Package</b> .                    |

**Method Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current method object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.         |



### 19.2.2.1.1.7 MethodConstraint Class

A *MethodConstraint* is a condition imposed on a method. It is accessed through either the Method *PreConditions* or Method *PostConditions* collection.

**Associated table in .EAP file:** *t\_operationpres* and *t\_operationposts*

#### **MethodConstraint Attributes:**

| Attribute         | Type                                                | Notes                                                                       |
|-------------------|-----------------------------------------------------|-----------------------------------------------------------------------------|
| <b>MethodID</b>   | <i>Long</i>                                         | Read/Write<br>The local ID of the associated method.                        |
| <b>Name</b>       | <i>String</i>                                       | Read/Write<br>The name of the constraint.                                   |
| <b>Notes</b>      | <i>String</i>                                       | Read/Write<br>Descriptive notes about this constraint.                      |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface. |
| <b>Type</b>       | <i>String</i>                                       | Read/Write<br>The constraint type.                                          |

#### **MethodConstraint Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                          |
|------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.          |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current <i>MethodConstraint</i> object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

### 19.2.2.1.1.8 MethodTag Class

A *MethodTag* is a Tagged Value associated with a method.

**Associated table in .EAP file:** *t\_operationtag*

#### **MethodTag Attributes:**

| Attribute         | Type                                                | Notes                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FQName</b>     | <i>String</i>                                       | Read only<br>The fully-qualified name of the tag.                                                                                                                                                                                                                                                                                                                                                                              |
| <b>MethodID</b>   | <i>Long</i>                                         | Read/Write<br>The ID of the associated method.                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Name</b>       | <i>String</i>                                       | Read/Write<br>The tag or name of the property.                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Notes</b>      | <i>String</i>                                       | Read/Write<br>Further descriptive notes about this tag.<br>If <b>Value</b> (below) is set to " <b>&lt;memo&gt;</b> ", then <b>Notes</b> should contain the actual Tagged Value content.                                                                                                                                                                                                                                        |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                                                                                                                                                                                                                    |
| <b>TagGUID</b>    | <i>String</i>                                       | Read/Write<br>A unique GUID for this Tagged Value.                                                                                                                                                                                                                                                                                                                                                                             |
| <b>TagID</b>      | <i>Long</i>                                         | Read only<br>A unique ID for this Tagged Value.                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Value</b>      | <i>String</i>                                       | Read/Write<br>The value assigned to this tag.<br>This field has a 255 character limit. If the value is greater than 255 characters long, set the value to " <b>&lt;memo&gt;</b> " and insert the body of text in the <b>Notes</b> attribute (above).<br>When reading existing Tagged Values, if <b>Value</b> = " <b>&lt;memo&gt;</b> " then the developer should read the actual body of text from the <b>Notes</b> attribute. |

**MethodTag Methods:**

| Method                                                              | Type           | Notes                                                                                                                                                                                    |
|---------------------------------------------------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetAttribute</b><br>(string<br>propName)                         | <i>String</i>  | Returns the text of a single named property within a structured tagged value.                                                                                                            |
| <b>GetLastError ()</b>                                              | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.    |
| <b>HasAttributes()</b>                                              | <i>Boolean</i> | Returns true if the tagged values is a structured tagged value with one or more properties.                                                                                              |
| <b>SetAttribute</b><br>(string<br>propName,<br>string<br>propValue) | <i>Boolean</i> | Sets the text of a single named property within a structured tagged value.                                                                                                               |
| <b>Update ()</b>                                                    | <i>Boolean</i> | Updates the current <i>MethodTag</i> object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

**19.2.2.1.1.9 Parameter Class**

A Parameter object represents a method argument and is accessed through the *Method Parameters* collection.

**Associated table in .EAP file:** *t\_operationparams*

**Parameter Attributes:**

| Attribute           | Type          | Notes                                                         |
|---------------------|---------------|---------------------------------------------------------------|
| <b>Alias</b>        | <i>String</i> | Read/Write<br><br>An optional alias for this parameter.       |
| <b>ClassifierID</b> | <i>String</i> | Read/Write<br><br>A ClassifierID for the parameter, if known. |
| <b>Default</b>      | <i>String</i> | Read/Write<br><br>A default value for this parameter.         |

| Attribute            | Type                                              | Notes                                                                                         |
|----------------------|---------------------------------------------------|-----------------------------------------------------------------------------------------------|
| <b>IsConst</b>       | <i>Boolean</i>                                    | Read/Write<br>A flag indicating that the parameter is <i>Const</i> (cannot be altered).       |
| <b>Kind</b>          | <i>String</i>                                     | Read/Write<br>The parameter kind - <b>in</b> , <b>inout</b> , <b>out</b> , or <b>return</b> . |
| <b>Name</b>          | <i>String</i>                                     | Read/Write<br>The parameter name; this must be unique for a single method.                    |
| <b>Notes</b>         | <i>String</i>                                     | Read/Write<br>Descriptive notes.                                                              |
| <b>ObjectType</b>    | <a href="#">ObjectType</a><br><small>2419</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                   |
| <b>OperationID</b>   | <i>Long</i>                                       | Read only<br>The ID of the method associated with this parameter.                             |
| <b>ParameterGUID</b> | <i>String</i>                                     | Read/Write<br>A system generated, globally unique ID for the current Parameter.               |
| <b>Position</b>      | <i>Long</i>                                       | Read/Write<br>The position of the parameter in the argument list.                             |
| <b>Stereotype</b>    | <i>String</i>                                     | Read/Write<br>The first stereotype of the parameter.                                          |
| <b>StereotypeEx</b>  | <i>String</i>                                     | Read/Write<br>All the applied stereotypes of the parameter in a comma-separated list.         |
| <b>Style</b>         | <i>String</i>                                     | Read/Write<br>Some style information.                                                         |
| <b>StyleEx</b>       | <i>String</i>                                     | Read/Write                                                                                    |

| Attribute           | Type                                                                               | Notes                                                                                  |
|---------------------|------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
|                     |                                                                                    | Advanced style settings, reserved for the use of Sparx Systems.                        |
| <b>TaggedValues</b> | <a href="#">Collection</a><br>[2427] of type<br><a href="#">ParamTag</a><br>[2408] | Read/Write<br>The GUID of the parameter with which this <i>ParamTag</i> is associated. |
| <b>Type</b>         | <i>Variant</i>                                                                     | Read/Write<br>The parameter type; can be a primitive type or a defined classifier.     |

**Parameter Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current Parameter object after modifying or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.         |

**19.2.2.1.1.10 ParamTag Class**

A *ParamTag* is a Tagged Value associated with a method parameter.

**ParamTag Attributes:**

| Attribute          | Type                                 | Notes                                                                                  |
|--------------------|--------------------------------------|----------------------------------------------------------------------------------------|
| <b>ElementGUID</b> | <i>String</i>                        | Read/Write<br>The GUID of the parameter with which this <i>ParamTag</i> is associated. |
| <b>FQName</b>      | <i>String</i>                        | Read only<br>The fully qualified name of the tag.                                      |
| <b>ObjectType</b>  | <a href="#">ObjectType</a><br>[2419] | Read only<br>Distinguishes objects referenced through a Dispatch interface.            |

| Attribute           | Type          | Notes                                                               |
|---------------------|---------------|---------------------------------------------------------------------|
| <b>PropertyGUID</b> | <i>String</i> | Read/Write<br>A system generated GUID to identify the Tagged Value. |
| <b>Tag</b>          | <i>String</i> | Read/Write<br>The actual tag name.                                  |
| <b>Value</b>        | <i>String</i> | Read/Write<br>The value associated with this tag.                   |

**ParamTag Methods:**

| Method                                                 | Type           | Notes                                                                                                                                                                                 |
|--------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetAttribute(string propName)</b>                   | <i>String</i>  | Returns the text of a single named property within a structured tagged value.                                                                                                         |
| <b>GetLastError ()</b>                                 | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>HasAttributes()</b>                                 | <i>Boolean</i> | Returns true if the tagged values is a structured tagged value with one or more properties.                                                                                           |
| <b>SetAttribute(string propName, string propValue)</b> | <i>Boolean</i> | Sets the text of a single named property within a structured tagged value.                                                                                                            |
| <b>Update ()</b>                                       | <i>Boolean</i> | Updates the current <i>Param Tag</i> object after modifying or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

**19.2.2.1.1.11 Partitions Collection**

A collection of internal element partitions (regions). This is commonly seen in Activity, State, Boundary, Diagram Frame and similar elements. Not all elements support partitions.

This collection contains a set of *Partition* elements. The set is read/write: information is not saved until the host element is saved, so ensure that you call the *Element.Save* method after making changes to a *Partition*.

**Partition Attributes:**

| Attribute         | Type                                                | Notes                                                                                         |
|-------------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------|
| <b>Name</b>       | <i>String</i>                                       | Read/Write<br>The partition name; this can represent a condition or constraint in some cases. |
| <b>Note</b>       | <i>String</i>                                       | Read/Write<br>A free text note associated with this partition.                                |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                   |
| <b>Operator</b>   | <i>String</i>                                       | Read/Write<br>An optional operator value that specifies the partition type.                   |
| <b>Size</b>       | <i>String</i>                                       | Read/Write<br>The vertical or horizontal width of the partition in pixels.                    |

**Learn more**

- [Activity](#)[1101]
- [State](#)[1152]
- [Boundary](#)[1663]
- [Diagram Frame](#)[1119]

**19.2.2.1.1.12 Properties Class****Properties****Properties Attributes:**

| Attribute         | Type                                             | Notes                                                                       |
|-------------------|--------------------------------------------------|-----------------------------------------------------------------------------|
| <b>Count</b>      | <i>Long</i>                                      | The number of properties that are available for this object.                |
| <b>ObjectType</b> | <a href="#">ObjectType</a> <small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface. |

**Properties Methods:**

| Method                     | Type            | Notes                                                                                                                                                                                                                                                                                                               |
|----------------------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Item (object Index)</b> | <i>Property</i> | <p>Returns a property either by name or by a zero-based integer offset into the list of properties.</p> <p>Parameter:</p> <ul style="list-style-type: none"> <li>Index: Variant - either a string representing the property name or an integer representing the zero-based offset into the property list</li> </ul> |

**Property****Property Attributes:**

| Attribute         | Type                                         | Notes                                                                                                                                                                                                                                                                                                    |
|-------------------|----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>       | <i>String</i>                                | <p>Read only</p> <p>The name of the property.</p> <p>The object to which the properties list applies can have an automation property with the same name, in which case the data accessed through <b>Value</b> is identical to that obtained through the automation property.</p>                         |
| <b>ObjectType</b> | <a href="#">ObjectType</a> <sup>[2419]</sup> | <p>Read only</p> <p>Distinguishes objects referenced through a Dispatch interface.</p>                                                                                                                                                                                                                   |
| <b>Type</b>       | <a href="#">PropType</a> <sup>[2421]</sup>   | <p>Read only</p> <p>Provides an indication of what sort of data is going to be stored by this property. This restriction can be further defined by the <i>Validation</i> attribute.</p>                                                                                                                  |
| <b>Validation</b> | <i>String</i>                                | <p>Read only</p> <p>An optional string that is used to validate any data that is passed to the <i>Value</i> attribute. This string is used by the programmer at run time to provide an indication of what is expected, and by Enterprise Architect to ensure that the submitted data is appropriate.</p> |
| <b>Value</b>      | <i>Variant</i>                               | <p>Read/write</p> <p>The value of the property as defined in the other fields.</p>                                                                                                                                                                                                                       |



### 19.2.2.1.1.13 Transitions Collection

The Transitions collection applies only to *Timeline elements*.

A Timeline element displays **0** or more state transitions at set times on its extent. This collection enables you to access the transition set. You can also access additional information by referring to the connectors associated with the Timeline, and by referencing messages passed between timelines. Note that any changes made to elements in this collection are only saved when the main element is saved.

#### Transition Attributes:

| Attribute                 | Type                                                | Notes                                                                                                |
|---------------------------|-----------------------------------------------------|------------------------------------------------------------------------------------------------------|
| <b>DurationConstraint</b> | <i>String</i>                                       | Read/Write<br>A constraint on the time duration of the transition.                                   |
| <b>Event</b>              | <i>String</i>                                       | Read/Write<br>The event (optional) that initiated the transition.                                    |
| <b>Note</b>               | <i>String</i>                                       | Read/Write<br>A free text note.                                                                      |
| <b>ObjectType</b>         | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                          |
| <b>TimeConstraint</b>     | <i>String</i>                                       | Read/Write<br>A constraint on when the transition has to be completed.                               |
| <b>TxState</b>            | <i>String</i>                                       | Read/Writ<br>The state to transition to, as defined in the Timeline Properties dialog.               |
| <b>TxTime</b>             | <i>String</i>                                       | Read/Write.<br>The time that the transition occurs. The value depends on a range set in the diagram. |

### 19.2.2.2 App Object

The *App* object represents a running instance of Enterprise Architect. Its object provides access to the Automation Interface.




| Attribute | Type    | Notes     |
|-----------|---------|-----------|
| Project   | Project | Read only |

| Attribute  | Type       | Notes                                                    |
|------------|------------|----------------------------------------------------------|
|            |            | Provides a handle to the Project Interface.              |
| Repository | Repository | Read only<br>Provides a handle to the Repository object. |
| Visible    | Boolean    | Read/Write<br>Whether or not the application is visible. |

| Topic                      | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Link |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| <b>GetObject() Support</b> | <p>The <i>App</i> object is creatable and a handle can be obtained by creating one. In addition, clients can use the equivalent of Visual Basic's <i>GetObject()</i> to obtain a reference to a currently running instance of Enterprise Architect.</p> <p>Use this method to more quickly test changes to Add-Ins and external clients, as the Enterprise Architect application and data files do not have to be constantly re-loaded.</p> <p>For example:</p> <pre>Dim App as EA.App Set App = GetObject(, "EA.App") MsgBox App.Repository.Models.Count</pre> <p>Another example, which uses the App object without saving it to a variable:</p> <pre>Dim Rep as EA.Repository Set Rep = GetObject(, "EA.App"). Repository MsgBox Rep.ConnectionString</pre> |      |

### 19.2.2.3 Enumerations

The following enumerations are defined by the Automation Interface:

| Enumeration            | Link                                                                                                                                     |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Constant Layout Styles | <a href="#">ConstLayoutStyles</a>  <sup>2414</sup>  |
| Create Baseline Flag   | <a href="#">CreateBaselineFlag</a>  <sup>2415</sup> |
| Create Model Type      | <a href="#">CreateModelType</a>  <sup>2416</sup>    |

| Enumeration                   | Link                                                   |
|-------------------------------|--------------------------------------------------------|
| Document Break                | <a href="#">DocumentBreak</a> <sup>[2416]</sup>        |
| Document Type                 | <a href="#">DocumentType</a> <sup>[2416]</sup>         |
| EA Edition Types              | <a href="#">EAEditionTypes</a> <sup>[2417]</sup>       |
| Enumeration Relation Set Type | <a href="#">EnumRelationSetType</a> <sup>[2417]</sup>  |
| Export Package XMI Flag       | <a href="#">ExportPackageXMIFlag</a> <sup>[2418]</sup> |
| Mail Interface Message Flag   | <a href="#">MessageFlag</a> <sup>[2419]</sup>          |
| MDG Menus                     | <a href="#">MDGMenus</a> <sup>[2418]</sup>             |
| Object Type                   | <a href="#">ObjectType</a> <sup>[2419]</sup>           |
| PropType                      | <a href="#">PropType</a> <sup>[2421]</sup>             |
| Reload Type                   | <a href="#">ReloadType</a> <sup>[2421]</sup>           |
| Scenario Diagram Type         | <a href="#">ScenarioDiagramType</a> <sup>[2422]</sup>  |
| Scenario Step Type            | <a href="#">ScenarioStepType</a> <sup>[2422]</sup>     |
| Scenario Test Type            | <a href="#">ScenarioTestType</a> <sup>[2423]</sup>     |
| XMI Type                      | <a href="#">XMIType</a> <sup>[2423]</sup>              |

### 19.2.2.3.1 ConstLayoutStyles

The *enum* values defined here are used exclusively for the *Lay Out a Diagram* method. They enable you to define the layout options as depicted in the **Diagram | Layout Diagram** menu option.

| Value                          | Meaning                                                                    |
|--------------------------------|----------------------------------------------------------------------------|
| <i>IsCrossReduceAggressive</i> | Perform aggressive Cross-reduction in the layout process (time consuming). |

| Value                              | Meaning                                                                      |
|------------------------------------|------------------------------------------------------------------------------|
|                                    |                                                                              |
| <i>IsCycleRemoveDFS</i>            | Use the <i>Depth First Cycle Removal</i> algorithm.                          |
| <i>IsCycleRemoveGreedy</i>         | Use the <i>Greedy Cycle Removal</i> algorithm.                               |
| <i>IsDiagramDefault</i>            | Use existing layout options specified for this diagram.                      |
| <i>IsInitializeDFSIn</i>           | Initialize the layout using the <i>Depth First Search Inward</i> algorithm.  |
| <i>IsInitializeNaive</i>           | Initialize the layout using the <i>Naïve Initialize Indices</i> algorithm.   |
| <i>IsInitializeDFSOut</i>          | Initialize the layout using the <i>Depth First Search Outward</i> algorithm. |
| <i>IsLayeringLongestPathSink</i>   | Layer the diagram using the <i>Longest Path Sink</i> algorithm.              |
| <i>IsLayeringLongestPathSource</i> | Layer the diagram using the <i>Longest Path Source</i> algorithm.            |
| <i>IsLayeringOptimalLinkLength</i> | Layer the diagram using the <i>Optimal Link Length</i> algorithm.            |
| <i>IsLayoutDirectionDown</i>       | Direct connectors to point down.                                             |
| <i>IsLayoutDirectionLeft</i>       | Direct connectors to point left.                                             |
| <i>IsLayoutDirectionRight</i>      | Direct connectors to point right.                                            |
| <i>IsLayoutDirectionUp</i>         | Direct connectors to point up.                                               |
| <i>IsProgramDefault</i>            | Use factory default layout options as specified by Enterprise Architect.     |

[Learn more](#)

- [Lay Out a Diagram](#) 

### 19.2.2.3.2 CreateBaselineFlag

The *CreateBaselineFlag* enumeration is used in Baseline Management, when creating a Baseline.

| Value               | Meaning                                                                                         |
|---------------------|-------------------------------------------------------------------------------------------------|
| <i>cbSaveToStub</i> | Baseline this package with only immediate children (child packages are included as stubs only). |

### 19.2.2.3.3 CreateModelType

The *CreateModelType* enumeration is used in the **CreateModel** method on the Repository Class.

| Value                         | Meaning                                                                                                              |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------|
| <i>cmEAPFromBase</i>          | Create a copy of the EABase model file to the specified file path.                                                   |
| <i>cmEAPFromSQLRepository</i> | Create a .eap file shortcut to an SQL-based repository; requires user interaction to provide sql connection details. |

#### Learn more

- [CreateModel Method](#) <sup>[2455]</sup>

### 19.2.2.3.4 DocumentBreak

The *DocumentBreak* enumeration is used in the **InsertBreak** method on the *DocumentGenerator* Class.

| Value               | Meaning                                 |
|---------------------|-----------------------------------------|
| <i>breakPage</i>    | Insert a page break in the document.    |
| <i>breakSection</i> | Insert a section break in the document. |

#### Learn more

- [InsertBreak Method](#) <sup>[2564]</sup>

### 19.2.2.3.5 DocumentType

The *DocumentType* enumeration is used in the **SaveDocument** method on the *DocumentGenerator* Class.

| Value         | Meaning                                             |
|---------------|-----------------------------------------------------|
| <i>dtHTML</i> | Save the document file to disk as an HTML document. |

| Value        | Meaning                                            |
|--------------|----------------------------------------------------|
| <i>dtRTF</i> | Save the document file to disk as an RTF document. |

#### Learn more

- [SaveDocument](#)<sup>[2565]</sup>

### 19.2.2.3.6 *EAEditionTypes*

#### Topics

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>General Usage</b> | <p>The <i>EAEditionTypes</i> enumeration identifies the current level of licensed functionality available</p> <pre> EAEditionTypes theEdition = theRepository. GetEAEdition(); if ( theEdition == EAEditionTypes.piDesktop )     ... else if ( theEdition == EAEditionTypes. piProfessional )     ... </pre> <p>The enumeration defines the following formal values:</p> <ul style="list-style-type: none"> <li>• <i>piLite</i></li> <li>• <i>piDesktop</i></li> <li>• <i>piProfessional</i></li> <li>• <i>piCorporate</i></li> <li>• <i>piBusiness</i></li> <li>• <i>piSystemEng</i></li> <li>• <i>piUltimate</i></li> </ul> <p>There is no separate value for the trial edition; the <i>Repository.GetEAEdition()</i> function returns the appropriate <b>EAEditionTypes</b> value for whichever edition the user has selected to trial.</p> |          |

### 19.2.2.3.7 *EnumRelationSetType*

This enumeration represents values returned from the *GetRelationSet* method of the **Element** object.

| Value              | Meaning                                              |
|--------------------|------------------------------------------------------|
| <i>rsDependEnd</i> | List of elements that depend on the current element. |

| Value                    | Meaning                                                       |
|--------------------------|---------------------------------------------------------------|
| <i>rsDependStart</i>     | List of elements that the current element depends on.         |
| <i>rsGeneralizeEnd</i>   | List of elements that are generalized by the current element. |
| <i>rsGeneralizeStart</i> | List of elements that the current element generalizes.        |
| <i>rsParents</i>         | List of all parent elements of the current element.           |
| <i>rsRealizeEnd</i>      | List of elements that are realized by the current element.    |
| <i>rsRealizeStart</i>    | List of elements that the current element realizes.           |

#### Learn more

- [Element Class](#) <sup>[2478]</sup>

#### 19.2.2.3.8 *ExportPackageXMIFlag*

The *ExportPackageXMIFlag* enumeration is used in package control, when exporting to XML.

| Value               | Meaning                                                                                       |
|---------------------|-----------------------------------------------------------------------------------------------|
| <i>epSaveToStub</i> | Export this package with only immediate children (child packages are included as stubs only). |

#### 19.2.2.3.9 *MDGMenus*

Use this enumeration when providing the *HiddenMenus* property to *MDG\_GetProperty*.

These options are exclusive of one another and can be read or added to hide more than one menu.

| Value                 | Meaning                                |
|-----------------------|----------------------------------------|
| <i>mgBuildProject</i> | Hide <b>Build Project</b> menu option. |
| <i>mgMerge</i>        | Hide <b>Merge</b> menu option.         |
| <i>mgRun</i>          | Hide <b>Run</b> menu option.           |

Learn more

- [MDG.GetProperty](#)<sup>[2678]</sup>

**19.2.2.3.10 MessageFlag**

The *MessageFlag* enumeration is used in both the **SendMailMessage** and **ComposeMailMessage** methods of the MailInterface, to specify a flag to attach to the message.

| Value             | Meaning                                |
|-------------------|----------------------------------------|
| <b>mfNone</b>     | Do not flag the message.               |
| <b>mfComplete</b> | Flag the message as 'Complete'.        |
| <b>mfPurple</b>   | Flag the message with a 'Purple' flag. |
| <b>mfOrange</b>   | Flag the message with a 'Orange' flag. |
| <b>mfGreen</b>    | Flag the message with a 'Green' flag.  |
| <b>mfYellow</b>   | Flag the message with a 'Yellow' flag. |
| <b>mfBlue</b>     | Flag the message with a 'Blue' flag.   |
| <b>mfRed</b>      | Flag the message with a 'Red' flag.    |

**19.2.2.3.11 ObjectType**

| Topic                           | Detail                                                                                                                                                                                                                                                                                            | See also |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>General Usage</b>            | <p>The <i>ObjectType</i> enumeration identifies Enterprise Architect object types even when referenced through a Dispatch interface</p> <p>For example:</p> <pre> Object ob = Repository.GetElementById(13); if ( ob.ObjectType == otElement ) ; else if ( ob.ObjectType == otAuthor ) ... </pre> |          |
| <b>Valid Enumeration Values</b> | <p><i>otAttribute</i></p> <p><i>otAttributeConstraint</i></p>                                                                                                                                                                                                                                     |          |



| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See also |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <i>otAttributeTag</i><br><i>otAuthor</i><br><i>otClient</i><br><i>otCollection</i><br><i>otConnector</i><br><i>otConnectorConstraint</i><br><i>otConnectorEnd</i><br><i>otConnectorTag</i><br><i>otConstraint</i><br><i>otCustomProperty</i><br><i>otDatatype</i><br><i>otDiagram</i><br><i>otDiagramLink</i><br><i>otDiagramObject</i><br><i>otEffort</i><br><i>otElement</i><br><i>otEventProperties</i><br><i>otEventProperty</i><br><i>otFile</i><br><i>otIssue</i><br><i>otMailInterface</i><br><i>otMethod</i><br><i>otMethodConstraint</i><br><i>otMethodTag</i><br><i>otMetric</i><br><i>otModel</i><br><i>otNone</i><br><i>otPackage</i><br><i>otParameter</i><br><i>otParamTag</i><br><i>otPartition</i><br><i>otProject</i><br><i>otProjectIssues</i><br><i>otProjectResource</i><br><i>otProperties</i><br><i>otProperty</i><br><i>otPropertyType</i><br><i>otReference</i><br><i>otRepository</i><br><i>otRequirement</i><br><i>otResource</i><br><i>otRisk</i><br><i>otRoleTag</i><br><i>otScenario</i><br><i>otScenarioExtension</i><br><i>otScenarioStep</i><br><i>otStereotype</i><br><i>otSwimlane</i><br><i>otSwimlaneDef</i><br><i>otSwimlanes</i><br><i>otTaggedValue</i><br><i>otTask</i><br><i>otTerm</i><br><i>otTest</i><br><i>otTransition</i> |          |

### 19.2.2.3.12 PropType

The *PropType* enumeration gives the automation programmer an indication of what sort of data is going to be stored by this property.

| Value                  | Meaning                                                                                                     |
|------------------------|-------------------------------------------------------------------------------------------------------------|
| <i>ptArray</i>         | An array containing values of any type.                                                                     |
| <i>ptBoolean</i>       | <b>True</b> or <b>False</b> .                                                                               |
| <i>ptEnum</i>          | A string being an entry in the semi-colon separated list specified in the validation field of the Property. |
| <i>ptFloatingPoint</i> | 4 or 8 byte floating point value.                                                                           |
| <i>ptInteger</i>       | 16-bit or 32-bit signed integer.                                                                            |
| <i>ptString</i>        | Unicode string.                                                                                             |

### 19.2.2.3.13 ReloadType

This enumeration represents values returned from the *GetReloadItem* and *PeekReloadItem* methods of the *ModelWatcher* Class. It has four possible values, which define the type of change that was made to a model.

| Value                | Meaning                                                                          |
|----------------------|----------------------------------------------------------------------------------|
| <i>rtElement</i>     | The <i>Item</i> parameter represents a particular element that must be reloaded. |
| <i>rtEntireModel</i> | Entire model must be reloaded to ensure that all changes are reloaded.           |
| <i>rtNone</i>        | No change in the model.                                                          |
| <i>rtPackage</i>     | The <i>Item</i> parameter represents a particular package that must be reloaded. |

### 19.2.2.3.14 ScenarioDiagramType

The *ScenarioDiagramType* enumeration provides the following enumeration values to the **Project.GenerateDiagramFromScenario()** method. They specify the type of diagram to generate.

| Value                                  | Meaning                                                 | See also                                                         |
|----------------------------------------|---------------------------------------------------------|------------------------------------------------------------------|
| <i>sdActivity</i>                      | Generate an Activity diagram.                           | <a href="#">Generated Activity diagram</a> <sup>[865]</sup>      |
| <i>sdActivityWithAction</i>            | Generate an Activity diagram with an Action.            | <a href="#">Generated Activity diagram</a> <sup>[865]</sup>      |
| <i>sdActivityWithActionPin</i>         | Generate an Activity diagram with an ActionPin.         | <a href="#">Generated Activity diagram</a> <sup>[865]</sup>      |
| <i>sdActivityWithActivityParameter</i> | Generate an Activity diagram with an ActivityParameter. | <a href="#">Generated Activity diagram</a> <sup>[865]</sup>      |
| <i>sdRobustness</i>                    | Generate a Robustness diagram.                          | <a href="#">Generated Robustness diagram</a> <sup>[871]</sup>    |
| <i>sdRuleFlow</i>                      | Generate a RuleFlow diagram                             | <a href="#">Generated RuleFlow diagram</a> <sup>[867]</sup>      |
| <i>sdSequence</i>                      | Generate a Sequence diagram.                            | <a href="#">Generated Sequence diagram</a> <sup>[869]</sup>      |
| <i>sdState</i>                         | Generate a State Machine diagram.                       | <a href="#">Generated State Machine diagram</a> <sup>[867]</sup> |

#### Learn more

- [Project.GenerateDiagramFromScenario\(\) Method](#) <sup>[2543]</sup>

### 19.2.2.3.15 ScenarioStepType

The *ScenarioStepType* enumeration is used to identify the steps of a scenario, and the entity performing the step.

| Value           | Meaning                                                      | See also |
|-----------------|--------------------------------------------------------------|----------|
| <i>stActor</i>  | Identify that the step is an action performed by an actor.   |          |
| <i>stSystem</i> | Identify that the step is an action performed by the system. |          |

#### Learn more

- [Scenario Step Class](#) <sup>[2502]</sup>

#### 19.2.2.3.16 ScenarioTestType

The *ScenarioTestType* enumeration provides the following enumeration values to the **Project.GenerateTestFromScenario()** method. They specify the type of test to generate.

| Value             | Meaning                                 | See also |
|-------------------|-----------------------------------------|----------|
| <i>stExternal</i> | Generate an external Test Case element. |          |
| <i>stInternal</i> | Generate an internal test.              |          |

#### Learn more

- [Project.GenerateTestFromScenario\(\) Method](#) <sup>[2546]</sup>
- [Generate Test Classes](#) <sup>[873]</sup>

#### 19.2.2.3.17 XMIType

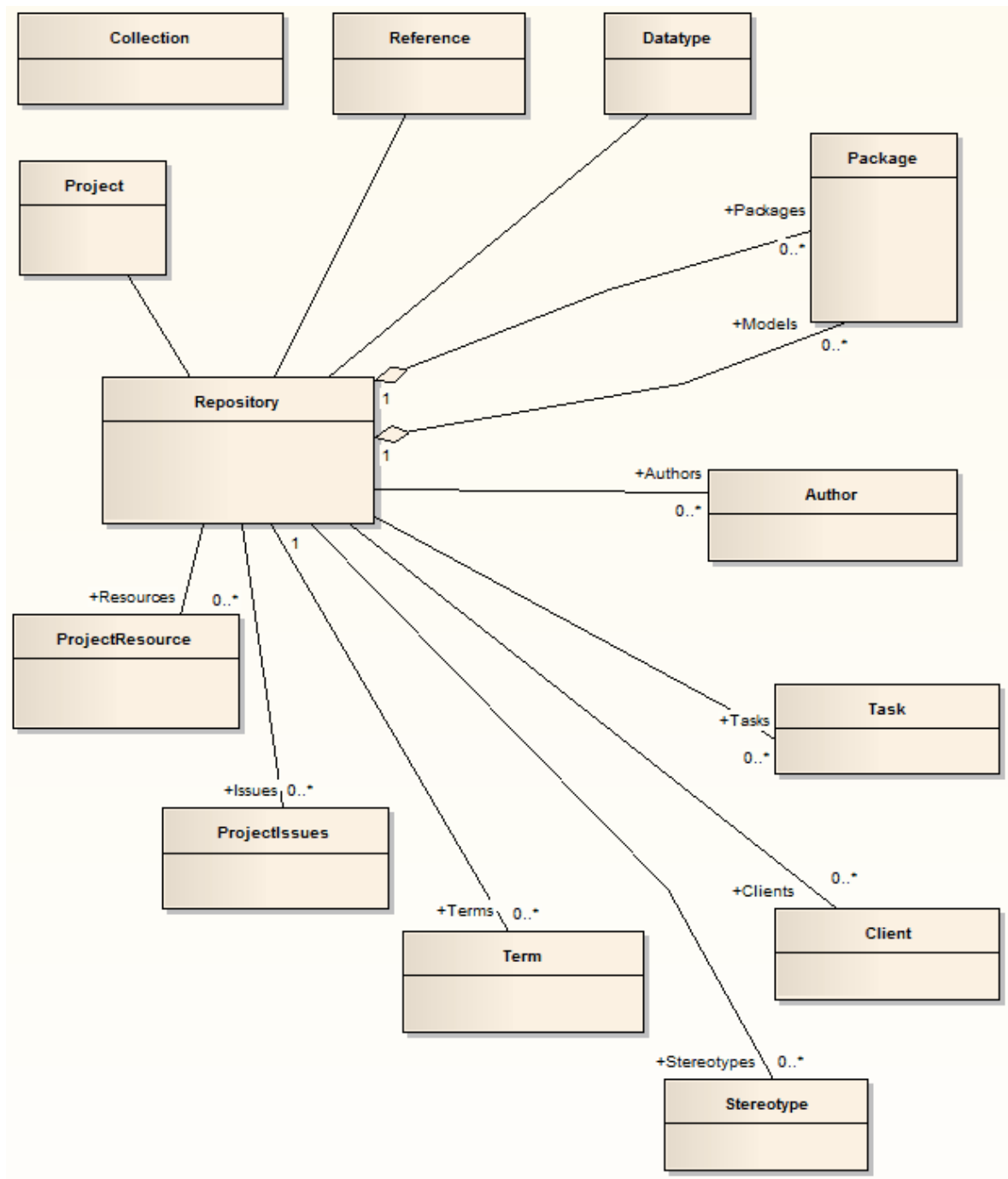
The following enumeration values are used in the *Project.ExportPackageXML()* method. They enable specification of the XML export type.

*xmiEADefault*  
*xmiRoseDefault*  
*xmiEA10*  
*xmiEA11*  
*xmiEA12*  
*xmiEA20*  
*xmiEA21*  
*xmiMOF13*  
*xmiMOF14*  
*xmiRose10*  
*xmiRose11*  
*xmiRose12*

#### 19.2.2.4 Repository Package

The *Repository* package contains the high level system objects and entry point into the model itself using the *Models* collection and the other system level collections.

This diagram illustrates the **Repository** and its first level functions and collections.



#### Learn more

- [Repository Class](#) <sup>2448</sup>

#### Learning Center topics

- (Alt+F1) | [Automation](#) | [Automation](#) | [Introduction to Automation](#)

### 19.2.2.4.1 Author Class

An *Author* object represents a named model author. Authors can be accessed using the Repository *Authors* collection.

**Associated table in .EAP file:** *t\_authors*

#### Author Attributes:

| Attribute         | Type                                 | Notes                                                                       |
|-------------------|--------------------------------------|-----------------------------------------------------------------------------|
| <b>Name</b>       | <i>String</i>                        | Read/Write<br>The Author name.                                              |
| <b>Notes</b>      | <i>String</i>                        | Read/Write<br>Notes about the author.                                       |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br>[2419] | Read only<br>Distinguishes objects referenced through a Dispatch interface. |
| <b>Roles</b>      | <i>String</i>                        | Read/Write<br>Roles the author might play in this project.                  |

#### Author Methods:

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current Author object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.               |

### 19.2.2.4.2 Client Class

A Client represents one or more people or organizations related to the project. Clients can be accessed using the Repository *Clients* collection.

**Associated table in .EAP file:** *t\_clients*

**Client Attributes:**

| Attribute           | Type                                                | Notes                                                                                |
|---------------------|-----------------------------------------------------|--------------------------------------------------------------------------------------|
| <b>EEmail</b>       | <i>String</i>                                       | Read/Write<br>The client's email address.                                            |
| <b>Fax</b>          | <i>String</i>                                       | Read/Write<br>The client's fax number.                                               |
| <b>Mobile</b>       | <i>String</i>                                       | Read/Write<br>The client's mobile phone number, if available.                        |
| <b>Name</b>         | <i>String</i>                                       | Read/Write<br>The client's name.                                                     |
| <b>Notes</b>        | <i>String</i>                                       | Read/Write<br>Notes about the client.                                                |
| <b>ObjectType</b>   | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through the <i>Dispatch</i> interface. |
| <b>Organization</b> | <i>String</i>                                       | Read/Write<br>The client's associated organization.                                  |
| <b>Phone1</b>       | <i>String</i>                                       | Read/Write<br>The client's main phone number.                                        |
| <b>Phone2</b>       | <i>String</i>                                       | Read/Write<br>The client's second phone number.                                      |
| <b>Roles</b>        | <i>String</i>                                       | Read/Write<br>Roles this client might play in the project.                           |

**Client Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current Client object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.        |

#### 19.2.2.4.3 Collection Class

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Link |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| <b>General Usage</b> | <p>This is the main collection Class used by all elements within the Automation Interface. It contains methods to iterate through the collection, refresh the collection and delete an item from the collection. It is important to realize that when <i>AddNew</i> is called, the item is not automatically added to the current collection.</p> <p>The typical steps are:</p> <ul style="list-style-type: none"> <li>• Call <i>AddNew</i> to add a new item</li> <li>• Modify the item as required</li> <li>• Call <i>Update</i> on the item to save it to the database</li> <li>• Call <i>Refresh</i> on the collection to include it in the current set</li> </ul> <p>Delete is much the same; until <i>Refresh</i> is called, the collection still contains a reference to the deleted item, which should not be called.</p> <p>Each can be used to iterate through the collection for languages that support this type of construct.</p> |      |

#### Collection Attributes:

| Attribute         | Type                                                | Notes                                                                       |
|-------------------|-----------------------------------------------------|-----------------------------------------------------------------------------|
| <b>Count</b>      | <i>Short</i>                                        | Read only<br>The number of objects referenced by this list.                 |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface. |



| Attribute | Type | Notes |
|-----------|------|-------|
|           |      |       |

**Collection Methods:**

| Method                                         | Type          | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | See also                                                                                                                                                                                                                        |
|------------------------------------------------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AddNew (string Name, string Type)</b>       | <i>Object</i> | <p>Adds a new item to the current collection.</p> <p>Note that the interface is the same for all collections; you must provide a <i>Name</i> and <i>Type</i> argument. What these are used for depends on the actual collection member.</p> <p>Also note that you must call <i>Update()</i> on the returned object to complete the <i>AddNew</i>. If <i>Update()</i> is not called the object is left in an indeterminate state.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Name: String</li> <li>• Type: String (up to 30 characters long)</li> </ul> |                                                                                                                                                                                                                                 |
| <b>Delete (short index)</b>                    | <i>Void</i>   | <p>Deletes the item at the selected reference.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• index: Short</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                 |
| <b>DeleteAt (short index, boolean Refresh)</b> | <i>Void</i>   | <p>Deletes the item at the selected index. The second parameter is currently unused.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• index: Short</li> <li>• Refresh: Boolean</li> </ul>                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                 |
| <b>GetAt (short index)</b>                     | <i>Object</i> | <p>Retrieves the array object using a numerical index. If the index is out of bounds, an error occurs.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• index: Short</li> </ul>                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                 |
| <b>GetByName (string Name)</b>                 | <i>Object</i> | <p>Gets an item in the current collection by Name.</p> <p>If the collection does not contain any items, the method returns a null value. If the collection contains items, but it was unable to find an object with the specified name, the method raises an exception.</p> <p>Only supported for the following collections: <i>Model</i>, <i>Package</i>, <i>Element</i>, <i>Diagram</i>, and element <i>TaggedValue</i>.</p>                                                                                                                                              | <p><a href="#">Package Class</a> <sup>[2433]</sup></p> <p><a href="#">Element Class</a> <sup>[2478]</sup></p> <p><a href="#">Diagram Class</a> <sup>[2523]</sup></p> <p><a href="#">TaggedValue Class</a> <sup>[2504]</sup></p> |

| Method                 | Type          | Notes                                                                                                                                                                                 | See also |
|------------------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                        |               | Parameters: <ul style="list-style-type: none"> <li>Name: String</li> </ul>                                                                                                            |          |
| <b>GetLastError ()</b> | <i>String</i> | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |          |
| <b>Refresh ()</b>      | <i>Void</i>   | Refreshes the collection by re-querying the model and reloading the collection. Should be called after adding a new item or after deleting an item.                                   |          |

#### 19.2.2.4.4 Datatype Class

A *Datatype* is a named type that can be associated with attribute or method types. It typically is related to either code engineering or database modeling. Datatypes also indicate which language or database system they relate to. Datatypes can be accessed using the Repository *Datatypes* collection.

**Associated table in .EAP file:** *t\_datatypes*

#### Datatype Attributes:

| Attribute           | Type          | Notes                                                                                                    |
|---------------------|---------------|----------------------------------------------------------------------------------------------------------|
| <b>DatatypeID</b>   | <i>Long</i>   | Read/Write<br><br>The instance ID for this datatype within the current model; this is system maintained. |
| <b>DefaultLen</b>   | <i>Long</i>   | Read/Write<br><br>The default length (DDL only).                                                         |
| <b>DefaultPrec</b>  | <i>Long</i>   | Read/Write<br><br>The default precision (DDL only).                                                      |
| <b>DefaultScale</b> | <i>Long</i>   | Read/Write<br><br>The default scale (DDL only).                                                          |
| <b>GenericType</b>  | <i>String</i> | Read/Write<br><br>The associated generic type for this data type.                                        |

| Attribute          | Type                                               | Notes                                                                                                                                                                                                                                  |
|--------------------|----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>HasLength</b>   | <i>String</i>                                      | Read/Write<br>Indicates whether the datatype has a length component.                                                                                                                                                                   |
| <b>MaxLen</b>      | <i>Long</i>                                        | Read/Write<br>The maximum length (DDL only).                                                                                                                                                                                           |
| <b>MaxPrec</b>     | <i>Long</i>                                        | Read/Write<br>The maximum precision (DDL only).                                                                                                                                                                                        |
| <b>MaxScale</b>    | <i>Long</i>                                        | Read/Write<br>The maximum scale (DDL only).                                                                                                                                                                                            |
| <b>Name</b>        | <i>String</i>                                      | Read/Write<br>The datatype name (such as <i>integer</i> ). This appears in the related drop-down datatype lists where appropriate.                                                                                                     |
| <b>ObjectType</b>  | <a href="#">ObjectType</a><br><small>[241]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                            |
| <b>Product</b>     | <i>String</i>                                      | Read/Write<br>The datatype product, such as Java, C++ or Oracle.                                                                                                                                                                       |
| <b>Size</b>        | <i>Long</i>                                        | Read/Write<br>The datatype size.                                                                                                                                                                                                       |
| <b>Type</b>        | <i>String</i>                                      | Read/Write<br>The type can be <i>DDL</i> for database datatypes or <i>Code</i> for language datatypes.                                                                                                                                 |
| <b>UserDefined</b> | <i>Long</i>                                        | Read/Write<br>Indicates if the datatype is a user defined type or system generated.<br>Datatypes distributed with Enterprise Architect are all system generated. Datatypes created in the Datatype dialog are marked <b>1 (true)</b> . |

### Datatype Methods

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current Datatype object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.      |

#### 19.2.2.4.5 EventProperties Class

An *EventProperties* object is passed to *BroadcastFunctions* to facilitate parameter passing.

##### EventProperties Attributes:

| Attribute         | Type                                         | Notes                                                                           |
|-------------------|----------------------------------------------|---------------------------------------------------------------------------------|
| <b>Count</b>      | <i>Long</i>                                  | Read only<br><br>The number of parameters being passed to this broadcast event. |
| <b>ObjectType</b> | <a href="#">ObjectType</a> <sup>[2419]</sup> | Read only<br><br>Distinguishes objects referenced through a Dispatch interface. |

##### EventProperties Methods:

| Method                    | Type                                            | Notes                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Get (object Index)</b> | <a href="#">EventProperty</a> <sup>[2432]</sup> | Read only<br><br>Returns an <i>EventProperty</i> in the list, raising an error if <i>Index</i> is out of range.<br><br>Parameters: <ul style="list-style-type: none"> <li>Index: Variant - can either be a number representing a zero-based index into the array, or a string representing the name of the <i>EventProperty</i>; for example, <i>Props.Get(3)</i> or <i>Props.Get("ObjectID")</i></li> </ul> |

#### 19.2.2.4.6 EventProperty Class

*EventProperty* objects are always part of an *EventProperties* collection, and are passed to Add-In methods responding to broadcast events.

##### EventProperty Attributes:

| Attribute          | Type                                 | Notes                                                                 |
|--------------------|--------------------------------------|-----------------------------------------------------------------------|
| <b>Description</b> | <i>String</i>                        | An explanation of what this property represents.                      |
| <b>Name</b>        | <i>String</i>                        | A string distinguishing this property from others in the list.        |
| <b>ObjectType</b>  | <a href="#">ObjectType</a><br>[2419] | Distinguishes objects referenced through a Dispatch interface.        |
| <b>Value</b>       | <i>VARIANT</i>                       | A string, number or object reference representing the property value. |

##### Learn more

- [EventProperties Class](#) [2431]
- [Broadcast Events](#) [2604]

#### 19.2.2.4.7 ModelWatcher Class

The *ModelWatcher* object enables an automation client to track changes in a particular model.

##### ModelWatcher Attributes:

| Attribute         | Type                                 | Notes                                                                       |
|-------------------|--------------------------------------|-----------------------------------------------------------------------------|
| <b>ObjectType</b> | <a href="#">ObjectType</a><br>[2419] | Read only<br>Distinguishes objects referenced through a Dispatch interface. |

##### ModelWatcher Methods:

| Methods                            | Type                                 | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------------|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetReloadItem (object Item)</b> | <a href="#">ReloadType</a><br>[2421] | The object that must be reloaded in order to see all changes is returned through the <i>Item</i> parameter. If there are no changes or the entire model must be reloaded, this value is returned as <b>null</b> (C#) or <b>Nothing</b> (VB).<br><br>Calling this method clears the records so that the next time it is called the return values refer only to new changes.<br><br>Returns a value from the <i>ReloadType</i> enumeration that specifies which |

| Methods               | Type                                               | Notes                                                                                                                          |
|-----------------------|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
|                       |                                                    | <p>type of change, if any, has occurred.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Item: Object</li> </ul> |
| <b>PeekReloadItem</b> | <a href="#">ReloadType</a><br><small>[242]</small> | This method behaves identically to <i>GetReloadItem()</i> but does not clear the change record.                                |

### Notes

- After your model has been loaded, you only create the ModelWatcher once; if you **reload** the model, or load another model, the created ModelWatcher is still valid

### Learn more

- [Refresh View of Shared Project](#)[242]

#### 19.2.2.4.8 Package Class

A *Package* object corresponds to a Package element in the Enterprise Architect Project Browser. Packages can be accessed either through the *Repository Models* collection (a Model is a special form of Package) or through the Package *Packages* collection.

Note that a Package has an Element object as an attribute; this corresponds to an Enterprise Architect Package element in the *t\_object* table and is used to associate additional information (such as scenarios and constraints) with the logical package.

To set additional information for a package, reference the Element object directly. Also note that if you add a Package to a diagram, you should add an instance of the element (not the Package itself) to the *Diagram Object Class* for a diagram.

**Associated table in .EAP file:** *t\_package*

### Package Attributes:

| Attribute        | Type          | Notes                                                                                                                                                      |
|------------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Alias</b>     | <i>String</i> | <p>Read only</p> <p>Alias</p>                                                                                                                              |
| <b>BatchLoad</b> | <i>Long</i>   | <p>Read/Write</p> <p>Flag to indicate that the package is batch loaded during batch import from controlled packages.</p> <p><b>Not currently used.</b></p> |

| Attribute           | Type                                                | Notes                                                                                                                                                                  |
|---------------------|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>BatchSave</b>    | <i>Long</i>                                         | Read/Write<br>Boolean value to indicate whether the package is included in the batch XML export list or not.                                                           |
| <b>CodePath</b>     | <i>String</i>                                       | Read/Write<br>The path where associated source code is found.<br><b>Not currently used.</b>                                                                            |
| <b>Connectors</b>   | <a href="#">Collection</a><br><small>[2427]</small> | Read only<br>The collection of connectors.                                                                                                                             |
| <b>Created</b>      | <i>Date</i>                                         | Read/Write<br>Date the package was created.                                                                                                                            |
| <b>Diagrams</b>     | <a href="#">Collection</a><br><small>[2427]</small> | Read only<br>A collection of diagrams contained in this package.                                                                                                       |
| <b>Element</b>      | <a href="#">Element</a><br><small>[2478]</small>    | Read only<br>The associated element object; use to get/set common information such as Stereotype, Complexity, Alias, Author, Constraints, Tagged Values and Scenarios. |
| <b>Elements</b>     | <a href="#">Collection</a><br><small>[2427]</small> | Read only<br>A collection of elements that belong to this package.                                                                                                     |
| <b>Flags</b>        | <i>String</i>                                       | Read/Write<br>Extended information about the package.                                                                                                                  |
| <b>IsControlled</b> | <i>Boolean</i>                                      | Read/Write<br>Indicates if the package has been marked as <i>Controlled</i> .                                                                                          |
| <b>IsModel</b>      | <i>Boolean</i>                                      | Read only<br>Indicates if the package is a model or a package.                                                                                                         |
| <b>IsNamespace</b>  | <i>Boolean</i>                                      | Read/Write<br><b>True</b> indicates that 'package is a Namespace root'.                                                                                                |

| Attribute                  | Type                                                | Notes                                                                        |
|----------------------------|-----------------------------------------------------|------------------------------------------------------------------------------|
|                            |                                                     | Use <b>0</b> and <b>1</b> to set <b>False</b> and <b>True</b> .              |
| <b>IsProtected</b>         | <i>Boolean</i>                                      | Read/Write<br>Indicates if the package has been marked as <i>Protected</i> . |
| <b>IsVersionControlled</b> | <i>Boolean</i>                                      | Read only<br>Indicates whether or not this package is under version control. |
| <b>LastLoadDate</b>        | <i>Date</i>                                         | Read/Write<br>The date XML was last loaded for the package.                  |
| <b>LastSaveDate</b>        | <i>Date</i>                                         | Read/Write<br>The date XML was last saved from the package.                  |
| <b>LogXML</b>              | <i>Boolean</i>                                      | Read/Write<br>Indicates if XML export information is to be logged.           |
| <b>Modified</b>            | <i>Date</i>                                         | Read/Write<br>Date the package was last modified.                            |
| <b>Name</b>                | <i>String</i>                                       | Read/Write<br>The name of the package.                                       |
| <b>Notes</b>               | <i>String</i>                                       | Read/Write<br>Notes about this package.                                      |
| <b>ObjectType</b>          | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.  |
| <b>Owner</b>               | <i>String</i>                                       | Read/Write.<br>The package owner when using controlled packages.             |
| <b>PackageGUID</b>         | <i>Variant</i>                                      | Read only<br>The global Package ID; valid across models.                     |



| Attribute         | Type                                              | Notes                                                                                                                                                 |
|-------------------|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PackageID</b>  | <i>Long</i>                                       | Read only<br>The local Package ID number.<br>Valid only in this model file.                                                                           |
| <b>Packages</b>   | <a href="#">Collection</a><br><small>2427</small> | Read only<br>A collection of contained packages that can be walked through.                                                                           |
| <b>ParentID</b>   | <i>Long</i>                                       | Read/Write<br>The ID of the package that is the parent of this one.<br><b>0</b> indicates this package is a <i>model</i> (that is, it has no parent). |
| <b>TreePos</b>    | <i>Long</i>                                       | Read/Write<br>The relative position in the tree compared to other packages (use to sort packages).                                                    |
| <b>UMLVersion</b> | <i>String</i>                                     | Read/Write<br>The UML version for XML export purposes.                                                                                                |
| <b>UseDTD</b>     | <i>Boolean</i>                                    | Read/Write<br>Indicates if a DTD is to be used when exporting XML.                                                                                    |
| <b>Version</b>    | <i>String</i>                                     | Read/Write<br>The version of the package.                                                                                                             |
| <b>XMLPath</b>    | <i>String</i>                                     | Read/Write<br>The path to which the XML is saved when using controlled packages.                                                                      |

**Package Methods:**

| Method                                    | Type           | Notes                                                                                                                                                                                                          |
|-------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ApplyGroupLock (string aGroupName)</b> | <i>Boolean</i> | Applies a group lock to the package object, for the specified group, on behalf of the current user.<br><br>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information |

| Method                                                        | Type                   | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------------------------------------|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                               |                        | Parameters: <ul style="list-style-type: none"> <li>aGroupName: String - The name of the security group for which to apply the lock</li> </ul>                                                                                                                                                                                                                                                                                                                                                                             |
| <b>ApplyUserLock ()</b>                                       | <i>Boolean</i>         | Applies a user lock to the package object for the current user.<br>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information.                                                                                                                                                                                                                                                                                                                                                   |
| <b>Clone</b>                                                  | <i>LDISPATC<br/>H</i>  | Inserts a copy of the package into the same parent as the original package.<br>Returns the newly-created package.                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>FindObject (string DottedID)</b>                           | <i>LPDISPAT<br/>CH</i> | Returns a package, element, attribute or operation matching the parameter <i>DottedID</i> .<br>If the <i>DottedID</i> is not found, an error is returned: <i>Can't find matching object</i> .<br>Parameters: <ul style="list-style-type: none"> <li>DottedID: String - Is in the form <i>object.object.object</i> where <i>object</i> is replaced by the name of a package, element attribute or operation; examples include <i>MyNamespace.Class1</i>, <i>CStudent.m_Name</i>, <i>MathClass.DoubleIt(int)</i></li> </ul> |
| <b>GetLastError ()</b>                                        | <i>String</i>          | Returns a string value describing the most recent error that occurred in relation to this object.<br>This function is rarely used as an exception is thrown when an error occurs.                                                                                                                                                                                                                                                                                                                                         |
| <b>ReleaseUserLock ()</b>                                     | <i>Boolean</i>         | Removes an existing User or Group lock from the package object.<br>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information.                                                                                                                                                                                                                                                                                                                                                   |
| <b>SetReadOnly (boolean ReadOnly, boolean IncludeSubPkgs)</b> | <i>Void</i>            | Sets a Package Flag to mark a package as <b>ReadOnly=1</b> .<br>If Project Security is enabled, the user must have <b>Configure Packages</b> permission to use this method.<br>Throws an exception if the operation fails due to the user not having <b>Configure Packages</b> permission; use <i>GetLastError()</i> to retrieve error information.<br>Parameters;                                                                                                                                                        |

| Method                                                                                               | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------------------------------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                      |                | <ul style="list-style-type: none"> <li>• <b>ReadOnly</b>: Boolean - Sets or clears the <i>Read Only</i> flag on the package(s); if: <ul style="list-style-type: none"> <li>• <b>False</b>, any <i>Read Only</i> flag is removed from the package</li> <li>• <b>True</b>, a <i>Read Only</i> flag is applied to the package</li> </ul> </li> <li>• <b>IncludeSubPkgs</b>: Boolean - Indicates whether to set/reset the <i>Read Only</i> flag on just the object package, or on the object package and all of the nested sub-packages that it contains; if: <ul style="list-style-type: none"> <li>• <b>False</b>, only the flag on the object package is set or cleared</li> <li>• <b>True</b>, flags are set (or cleared, according to the <i>ReadOnly</i> parameter) for the object package plus all of the nested sub-packages that it contains</li> </ul> </li> </ul> <p>When working with version controlled packages, the <i>Read Only</i> flag can be applied to packages whether they are checked-in or checked-out.</p> <p>User Security applies to setting this flag - if you are prevented from editing the package, you are also prevented from setting the flag.</p> |
| <b>Update ()</b>                                                                                     | <i>Boolean</i> | <p>Updates the current package object after modification or appending a new item.</p> <p>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.</p> <p>Note that a package object also has an <i>element</i> component that must be taken into account; the package object contains information about the package attributes such as hierarchy or contents.</p> <p>The element attribute contains information about, for example, Stereotype, Constraints or Files - all the attributes of a typical element.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>VersionControlAdd (string ConfigGuid, string XMLFile, string Comment, boolean KeepCheckedOut)</b> | <i>Void</i>    | <p>Places the package under version control, using the specified Version Control Configuration and the specified XML filename.</p> <p>Throws an exception if the operation fails; use <i>GetLastError() </i> to retrieve error information.</p> <p>It is recommended that the package be saved using <i>Update()</i> before calling <i>VersionControlAdd()</i>, so that any outstanding changes are not lost.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

| Method                                                                           | Type | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------------------------------------------------------------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                  |      | <p>Parameters:</p> <ul style="list-style-type: none"> <li>• ConfigGuid: String - Name corresponding to the Unique ID of the version control configuration to use</li> <li>• XMLFile: String - Name of the XML file to use for this package; this filename is relative to the Working Copy folder specified for the Config</li> <li>• Comment: String - Log message that is added to the version controlled file's history (where applicable)</li> <li>• KeepCheckedOut: Boolean - Specify <b>True</b> to add to version control and keep package checked-out</li> </ul>                                                                                                                                                                               |
| <b>VersionControlCheckin</b><br>(string Comment)                                 | Void | <p>Perform checkin of the version controlled package (also see <i>VersionControlCheckinEx</i>, below).</p> <p>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Comment: String - Log message that is added to the version controlled file's history (where applicable)</li> </ul>                                                                                                                                                                                                                                                                                                                                                 |
| <b>VersionControlCheckinEx</b><br>(string Comment, boolean PreserveCrossPkgRefs) | Void | <p>Perform checkin of the version controlled package.</p> <p>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Comment: String - Log message that is added to the version controlled file's history (where applicable)</li> <li>• PreserveCrossPkgRefs: Boolean - Flag to indicate whether to preserve or discard pre-existing Cross Package References when checking-in; this parameter overrides the setting in the Options dialog, XML Specifications page</li> </ul> <p>Unsatisfied cross package references are preserved or discarded according to this setting, without prompting the user; see <i>Learn more</i> below</p> |
| <b>VersionControlCheckout</b><br>(string Comment)                                | Void | <p>Perform checkout of the version controlled package.</p> <p>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information.</p> <p>Parameters:</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

| Method                                                         | Type        | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------------------------------------------------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                |             | <ul style="list-style-type: none"> <li>• Comment: String - Log message that is added to the version controlled file's history (where applicable)</li> </ul> <p>When working in an environment that uses a Private Model deployment and your model contains a significant number of cross-package references, it is recommended that you invoke the <i>Repository.ScanXMLAndReconcile()</i> method from time to time, following the re-importation of controlled packages - for example, after using <i>Package.VersionControlGetLatest()</i> to update a number of packages, or after performing a number of package check-outs.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>VersionControlGetLatest</b><br><b>(boolean ForceImport)</b> | <i>Void</i> | <p>Updates the local working copy of the package file associated with the object package, before re-importing the package data from the package file.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• ForceImport: Boolean - Used if the package data in the model is found to be up-to-date with respect to the version controlled package file; if: <ul style="list-style-type: none"> <li>• <b>False</b>, the package data that exists in the model is accepted as being up-to-date and no attempt is made to re-import data from the package file</li> <li>• <b>True</b>, Enterprise Architect re-imports the package from the package file regardless</li> </ul> </li> </ul> <p>See also the version control menu option <b>Get Latest</b>.</p> <p>When working in an environment that uses a Private Model deployment and your model contains a significant number of cross-package references, it is recommended that you invoke the <i>Repository.ScanXMLAndReconcile()</i> method from time to time, following the re-importation of controlled packages - for example, after using <i>Package.VersionControlGetLatest()</i> to update a number of packages, or after performing a number of package check-outs.</p> |
| <b>VersionControlGetStatus</b><br><b>()</b>                    | <i>Long</i> | <p>Returns the version control status of the package, as recorded in the current Enterprise Architect project database.</p> <p>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information.</p> <p>Return value maps to the following enumerated type:</p> <pre>enum EnumCheckOutStatus {     csUncontrolled = 0,     csCheckedIn,</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

| Method                                                           | Type        | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------------------------------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                  |             | <p><i>csCheckedOutToThisUser</i>,<br/> <i>csReadOnlyVersion</i>,<br/> <i>csCheckedOutToAnotherUser</i>,<br/> <i>csOfflineCheckedIn</i>,<br/> <i>csCheckedOutOfflineByUser</i>,<br/> <i>csCheckedOutOfflineByOther</i>,<br/> <i>csDeleted</i>,<br/> };</p> <p><i>csUncontrolled</i> - Either unable to communicate with the version control provider associated with the package or the package file is unknown to the provider.</p> <p><i>csCheckedIn</i> - The package is not checked-out to anybody in the current project database.</p> <p><i>csCheckedOutToThisUser</i> - The package is marked as checked-out to the current user, in the current project database.</p> <p><i>csReadOnlyVersion</i> - Package is marked as read-only; an earlier revision of the package has been retrieved from version control.</p> <p><i>csCheckedOutToAnotherUser</i> - The package is marked as checked-out in the current project database, by a user other than the current user.</p> <p><i>csOfflineCheckedIn</i> - The package is not checked-out to anybody in the current project database; however, the version control configuration associated with the package was unable to connect to the VC server.</p> <p><i>csCheckedOutOfflineByUser</i> - Indicates that the package was 'checked out' in this database, by this user, whilst disconnected from version control.</p> <p><i>csCheckedOutOfflineByOther</i> - Indicates that the package was checked out in this project database, by another user, whilst disconnected from version control.</p> <p><i>csDeleted</i> - The package file has been deleted from version control.</p> |
| <b>VersionControlPutLatest</b><br><b>(string CheckInComment)</b> | <i>Void</i> | <p>Perform a checkin of the version controlled package, whilst keeping the package checked-out.</p> <p>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information.</p> <p>When a package that was previously marked as <i>Checked Out Offline</i>, is successfully 'Put' (checkedin) to version control, that package's flags are updated to clear the <i>Checked Out Offline</i> indicator.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Comment: String - Log message added to the version controlled file's history (where applicable)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

| Method                                                            | Type        | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------------------------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VersionControlRemove ()</b>                                    | <i>Void</i> | Removes version control from the package.<br><br>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>VersionControlResynchPackageStatus (boolean ClearSettings)</b> |             | Synchronizes the version control status of the single object package recorded in your current model with the package status reported by your version control provider.<br><br>Parameters: <ul style="list-style-type: none"> <li>• <b>ClearSettings</b>: Boolean - used if the package file associated with the specified package is reported by the version control provider as uncontrolled; if <b>ClearSettings</b> is: <ul style="list-style-type: none"> <li>• <b>True</b>, the version control settings are cleared from the package</li> <li>• <b>False</b>, the version control settings remain unchanged</li> </ul> </li> </ul> |

**Learn more**

- [Repository Models](#)<sup>[2450]</sup>
- [Package Packages](#)<sup>[2438]</sup>
- [Version Control - Get Latest](#)<sup>[358]</sup>
- [Resynchronize the Status of Version Controlled Packages](#)<sup>[380]</sup>
- [Preserving Cross Package References](#)<sup>[433]</sup>

**19.2.2.4.9 ProjectIssues Class**

A system-level Issue that indicates a problem or risk associated with the system as a whole. ProjectIssues can be accessed using the Repository *Issues* collection.

**Associated table in .EAP file:** *t\_issues*

**ProjectIssues Attributes:**

| Attribute       | Type          | Notes                                                 |
|-----------------|---------------|-------------------------------------------------------|
| <b>Category</b> | <i>String</i> | Read/Write<br><br>The category this issue belongs to. |
| <b>Date</b>     | <i>Date</i>   | Read/Write                                            |

| Attribute           | Type                                                | Notes                                                                       |
|---------------------|-----------------------------------------------------|-----------------------------------------------------------------------------|
|                     |                                                     | The date the issue item was created.                                        |
| <b>DateResolved</b> | <i>Date</i>                                         | Read/Write<br>The date the issue was resolved.                              |
| <b>Name</b>         | <i>String</i>                                       | Read/Write<br>The issue name (that is, the issue itself).                   |
| <b>IssueID</b>      | <i>Long</i>                                         | Read only<br>The ID of this issue.                                          |
| <b>Notes</b>        | <i>String</i>                                       | Read/Write<br>The associated description of the issue.                      |
| <b>ObjectType</b>   | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface. |
| <b>Owner</b>        | <i>String</i>                                       | Read/Write<br>The owner of the issue.                                       |
| <b>Priority</b>     | <i>String</i>                                       | Read/Write<br>The issue priority - Low, Medium or High.                     |
| <b>Resolution</b>   | <i>String</i>                                       | Read/Write<br>A description of the resolution.                              |
| <b>Resolver</b>     | <i>String</i>                                       | Read/Write<br>The name of the person resolving the issue.                   |
| <b>Severity</b>     | <i>String</i>                                       | Read/Write<br>The issue severity - Low, Medium or High.                     |
| <b>Status</b>       | <i>String</i>                                       | Read/Write<br>The current status of the issue.                              |



**ProjectIssues Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current Issue object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.         |

**19.2.2.4.10 ProjectResource Class**

A *Project Resource* is a named person who is available to work on the current project in any capacity. *ProjectResources* can be accessed using the Repository *Resources* collection.

Associated table in .EAP file: *t\_resources*

**ProjectResource Attributes:**

| Attribute           | Type                                                       | Notes                                                                       |
|---------------------|------------------------------------------------------------|-----------------------------------------------------------------------------|
| <b>Email</b>        | <i>String</i>                                              | The resource's email address.                                               |
| <b>Fax</b>          | <i>String</i>                                              | The resource's fax number.                                                  |
| <b>Mobile</b>       | <i>Variant</i>                                             | The resource's mobile number, if available.                                 |
| <b>Name</b>         | <i>String</i>                                              | The name of the resource.                                                   |
| <b>Notes</b>        | <i>String</i>                                              | A description of the resource, if appropriate.                              |
| <b>ObjectType</b>   | <a href="#">ObjectType</a> <sup>[2419]</sup>               | Read only<br>Distinguishes objects referenced through a Dispatch interface. |
| <b>Organization</b> | <a href="#">Package</a> <sup>[2433]</sup><br><i>String</i> | The organization the resource is associated with.                           |
| <b>Phone1</b>       | <i>Variant</i>                                             | The resource's main telephone number.                                       |

| Attribute     | Type           | Notes                                                    |
|---------------|----------------|----------------------------------------------------------|
|               |                |                                                          |
| <b>Phone2</b> | <i>Variant</i> | The resource's alternative telephone number.             |
| <b>Roles</b>  | <i>String</i>  | The roles this resource can play in the current project. |

**ProjectResource Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                  |
|------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used, as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current Resource object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.       |

**19.2.2.4.11 ProjectRole Class**

A *ProjectRole* object represents a named project role. ProjectRoles can be accessed using the Repository *ProjectRole* collection.

**Associated table in .EAP file:** *t\_projectroles*

**ProjectRole Attributes:**

| Attribute          | Type                                                | Notes                                                                       |
|--------------------|-----------------------------------------------------|-----------------------------------------------------------------------------|
| <b>Description</b> | <i>String</i>                                       | Read/Write<br>The project role item description.                            |
| <b>Notes</b>       | <i>String</i>                                       | Read/Write<br>Notes about the project role item.                            |
| <b>ObjectType</b>  | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface. |

| Attribute   | Type          | Notes                                     |
|-------------|---------------|-------------------------------------------|
| <b>Role</b> | <i>String</i> | Read/Write<br>The project role item name. |

**ProjectRole Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current ProjectRole object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.   |

**19.2.2.4.12 PropertyType Class**

A *PropertyType* object represents a defined property that can be applied to UML elements as a Tagged Value. PropertyTypes can be accessed using the Repository *PropertyTypes* collection.

Each PropertyType corresponds to one of the predefined Tagged Values for the model.

**Associated table in .EAP file:** *t\_propertytypes*

**PropertyType Attributes:**

| Attribute          | Type                                         | Notes                                                                       |
|--------------------|----------------------------------------------|-----------------------------------------------------------------------------|
| <b>Description</b> | <i>String</i>                                | Read/Write<br>A short description of the property.                          |
| <b>Detail</b>      | <i>String</i>                                | Read/Write<br>Configuration information for the property.                   |
| <b>ObjectType</b>  | <a href="#">ObjectType</a> <sup>[2419]</sup> | Read only<br>Distinguishes objects referenced through a Dispatch interface. |
| <b>Tag</b>         | <i>String</i>                                | Read/Write<br>The name of the property (Tag Name).                          |

| Attribute | Type | Notes |
|-----------|------|-------|
|           |      |       |

**PropertyType Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                       |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.       |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current <i>PropertyType</i> object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

**19.2.2.4.13 Reference Class**

This Interface provides access to the various lookup tables within Enterprise Architect. Use the Repository *GetReferenceList()* method to get a handle to a list.

Valid lists are:

- Diagram
- Element
- Constraint
- Requirement
- Connector
- Status
- Cardinality
- Effort
- Metric
- Scenario
- Status
- Test

**Reference Attributes:**

| Attribute    | Type         | Notes                         |
|--------------|--------------|-------------------------------|
| <b>Count</b> | <i>Short</i> | A count of items in the list. |

| Attribute         | Type                                 | Notes                                                                       |
|-------------------|--------------------------------------|-----------------------------------------------------------------------------|
| <b>ObjectType</b> | <a href="#">ObjectType</a><br>[2419] | Read only<br>Distinguishes objects referenced through a Dispatch interface. |
| <b>Type</b>       | <i>String</i>                        | The list type (for example, <i>Diagram Types</i> ).                         |

**Reference Methods:**

| Method                     | Type          | Notes                                                                                                                                                                                 |
|----------------------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetAt (short Index)</b> | <i>String</i> | Get the item at the specified index.<br>Parameters: <ul style="list-style-type: none"> <li>Index: Short - The index of the item to retrieve from the list</li> </ul>                  |
| <b>GetLastError ()</b>     | <i>String</i> | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Refresh ()</b>          | <i>Short</i>  | Refresh the current list and return the count of items.                                                                                                                               |

**19.2.2.4.14 Repository Class**

The *Repository* is the main container of all structures such as models, packages and elements. You can iteratively begin accessing the model using the *Models* collection. It also has some convenient methods to directly access the structures without having to locate them in the hierarchy first.

**Associated table in .EAP file:** <none>

**Repository Attributes:**

| Attribute          | Type                                 | Notes                                                                                                                                                                                                                                                                                                   | See also |
|--------------------|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Authors</b>     | <a href="#">Collection</a><br>[2427] | Read only<br><br>This is the system <i>Authors</i> collection. This contains <b>0</b> or more <i>Author objects</i> , each of which can be associated with, for example, elements or diagrams as the item author or owner.<br><br>Use <i>AddNew</i> , <i>Delete</i> and <i>GetAt</i> to manage Authors. |          |
| <b>BatchAppend</b> | <i>Boolean</i>                       | Read/Write                                                                                                                                                                                                                                                                                              |          |

| Attribute               | Type                                                     | Notes                                                                                                                                                                                                                                                                                                                                           | See also |
|-------------------------|----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                         |                                                          | <p>Set this property to <b>true</b> when your automation client has to rapidly insert many elements, operations, attributes and/or operation parameters.</p> <p>Set to <b>false</b> when work is complete.</p> <p>This can result in 10- to 20-fold improvement in adding new elements in bulk.</p>                                             |          |
| <b>Clients</b>          | <a href="#">Collection</a><br><small>[2427]</small>      | <p>Read only</p> <p>A list of <i>Clients</i> associated with the project. You can modify, delete and add new <i>Client objects</i> using this collection.</p>                                                                                                                                                                                   |          |
| <b>ConnectionString</b> | <i>String</i>                                            | <p>Read only</p> <p>The filename/connection string of the current Repository.</p>                                                                                                                                                                                                                                                               |          |
| <b>Datatypes</b>        | <a href="#">Collection</a><br><small>[2427]</small>      | <p>Read only</p> <p>The <i>Datatypes</i> collection. This contains a list of <i>Datatype objects</i>, each representing a data type definition for either data modeling or code generation purposes.</p>                                                                                                                                        |          |
| <b>EAEdition</b>        | <a href="#">EAEdition Types</a><br><small>[2417]</small> | <p>Read only</p> <p>Returns the current level of core licensed functionality available.</p> <p>This property returns <b>Corporate</b> when the edition is <i>Business and Software Engineering</i>, <i>Systems Engineering</i> or <i>Ultimate</i>.</p> <p>Use <b>EAEditionEx</b> to identify which of these extended editions is available.</p> |          |
| <b>EAEditionEx</b>      | <a href="#">EAEdition Types</a><br><small>[2417]</small> | <p>Read only</p> <p>Returns the current level of extended licensed functionality available.</p>                                                                                                                                                                                                                                                 |          |
| <b>EnableCache</b>      | <i>Boolean</i>                                           | <p>Read/Write</p> <p>An optimization for pre-loading package objects when dealing with large sets of automation objects.</p>                                                                                                                                                                                                                    |          |
| <b>EnableUIUpdates</b>  | <i>Boolean</i>                                           | <p>Read/Write</p> <p>Set this property to <b>false</b> to improve the performance of changes to the model; for example, bulk addition of</p>                                                                                                                                                                                                    |          |

| Attribute                | Type                                                                                                               | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See also |
|--------------------------|--------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                          |                                                                                                                    | elements to a package. To reveal the changes to the user, call <i>Repository.RefreshModelView()</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |          |
| <b>FlagUpdate</b>        | <i>Boolean</i>                                                                                                     | Read/Write<br>Instructs Enterprise Architect to update the Repository with the <i>LastUpdate</i> value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |          |
| <b>InstanceGUID</b>      | <i>String</i>                                                                                                      | Read only<br>The identifier string identifying the Enterprise Architect runtime session.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |          |
| <b>IsSecurityEnabled</b> | <i>Boolean</i>                                                                                                     | Read only<br>Checks whether User Security is enabled for the current repository.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |          |
| <b>Issues</b>            | <a href="#">Collection</a><br><small>[2427]</small>                                                                | Read only<br>The <i>System Issues</i> list. Contains <i>ProjectIssues</i> objects, each detailing a particular issue as it relates to the project as a whole.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |          |
| <b>LastUpdate</b>        | <i>String</i>                                                                                                      | Read only<br>The identifier string identifying the Enterprise Architect runtime session and the timestamp for when it was set.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |          |
| <b>LibraryVersion</b>    | <i>Long</i>                                                                                                        | Read only<br>The build number of the Enterprise Architect runtime.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |          |
| <b>Models</b>            | <a href="#">Collection</a><br><small>[2427]</small><br>of type<br><a href="#">Package</a><br><small>[2433]</small> | Read only<br><i>Models</i> are of type <i>package</i> and belong to a collection of packages. This is the top level entry point to an Enterprise Architect project file. Each model is a <i>root node</i> in the Project Browser and can contain items such as Views and packages.<br><br>A model is a special form of a package; it has a <i>ParentID</i> of 0. By iterating through all models, you can access all the elements within the project hierarchy.<br><br>You can also use the <i>AddNew</i> function to create a new model. A model can be deleted, but remember that everything contained in the model is deleted as well. |          |
| <b>ObjectType</b>        | <a href="#">ObjectTyp</a>                                                                                          | Read only                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |          |

| Attribute                | Type                                         | Notes                                                                                                                                                                                                                                                                  | See also                                                                                                         |
|--------------------------|----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
|                          | <a href="#">e</a> <sup>[2419]</sup>          | Distinguishes objects referenced through the Dispatch interface.                                                                                                                                                                                                       |                                                                                                                  |
| <b>ProjectGUID</b>       | <i>String</i>                                | Read only<br>Returns a unique ID for the project.                                                                                                                                                                                                                      |                                                                                                                  |
| <b>ProjectRoles</b>      | <a href="#">Collection</a> <sup>[2427]</sup> | Read only<br>The system Roles collection. Contains <b>0</b> or more Role objects, each of which can be associated with, for example, elements or diagrams as the item author or owner.<br>Use <i>AddNew</i> , <i>Delete</i> and <i>GetAt</i> to manage Roles.          |                                                                                                                  |
| <b>PropertyTypes</b>     | <a href="#">Collection</a> <sup>[2427]</sup> | Read only<br>Collection of <b>Property Types</b> available to the Repository.                                                                                                                                                                                          | <a href="#">Property Type Class</a> <sup>[2446]</sup>                                                            |
| <b>Resources</b>         | <a href="#">Collection</a> <sup>[2427]</sup> | Read only<br>Contains available <i>ProjectResource</i> objects to assign to work items within the project.<br>Use the <b>add new</b> , <b>modify</b> and <b>delete</b> functions to manage resources.                                                                  |                                                                                                                  |
| <b>Stereotypes</b>       | <a href="#">Collection</a> <sup>[2427]</sup> | Read only<br>The <b>Stereotype</b> collection. A list of <i>Stereotype objects</i> that contain information on a stereotype and which elements it can be applied to.                                                                                                   | <a href="#">Stereotype Class</a> <sup>[2470]</sup>                                                               |
| <b>SuppressEADialogs</b> | <i>Boolean</i>                               | Read/Write<br>Set this property in the <i>EA_OnPostNewElement</i> or <i>EA_OnPostNewConnector</i> broadcast events to control whether Enterprise Architect should suppress showing the default Properties dialogs to the user when an element or connector is created. | <a href="#">EA_OnPostNewElement</a> <sup>[2635]</sup><br><a href="#">EA_OnPostNewConnector</a> <sup>[2636]</sup> |
| <b>Tasks</b>             | <a href="#">Collection</a> <sup>[2427]</sup> | Read only<br>A list of system tasks (to do list). Each entry is a <b>Task Item</b> ; you can modify, delete and add new tasks.                                                                                                                                         | <a href="#">Task Class</a> <sup>[2471]</sup>                                                                     |
| <b>Terms</b>             | <a href="#">Collection</a> <sup>[2427]</sup> | Read only<br>The project <i>Glossary</i> . Each <b>Term</b> object is an entry in                                                                                                                                                                                      | <a href="#">Term Class</a> <sup>[2473]</sup>                                                                     |



| Attribute | Type | Notes                                                                | See also |
|-----------|------|----------------------------------------------------------------------|----------|
|           |      | the Glossary. Add, modify and delete Terms to maintain the Glossary. |          |

**Repository Methods:**

| Method                                                | Type           | Notes                                                                                                                                                                                                                                                                                                                                 | See also                                           |
|-------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| <b>ActivateDiagram (Long DiagramID)</b>               |                | Activates an already open diagram (that is, makes it the active tab) in the main Enterprise Architect user interface.<br>Parameters:<br><ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to make active</li> </ul>                                                                                      |                                                    |
| <b>ActivatePerspective (string, long)</b>             | <i>Boolean</i> | <b>Deprecated</b> - no longer in use.                                                                                                                                                                                                                                                                                                 |                                                    |
| <b>ActivateTab (string Name)</b>                      |                | Activates an open Enterprise Architect tabbed view.<br>Parameters:<br><ul style="list-style-type: none"> <li>Name: String - the name of the view to activate</li> </ul>                                                                                                                                                               |                                                    |
| <b>ActivateTechnology (string Name)</b>               |                | Activates an enabled MDG Technology.<br>Parameters:<br><ul style="list-style-type: none"> <li>Name: String - the name of the Technology to activate</li> </ul>                                                                                                                                                                        |                                                    |
| <b>ActivateToolbox (string Toolbox, long Options)</b> | <i>Boolean</i> | Activates a Toolbox page in the GUI.<br>Returns <b>true</b> if the specified Toolbox page is successfully activated, otherwise returns <b>false</b> .<br>Parameters:<br><ul style="list-style-type: none"> <li>Toolbox: String - the name of the Toolbox page to activate</li> <li>Options: Long - reserved for future use</li> </ul> |                                                    |
| <b>AddDefinedSearches (string sXML)</b>               |                | Enables you to enter a set of defined searches that last in Enterprise Architect for the life of the application; when Enterprise Architect loads again they must be inserted again by your Add-In.                                                                                                                                   | <a href="#">Create and Manage Searches</a><br>1621 |

| Method                                                   | Type                          | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See also                                |
|----------------------------------------------------------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
|                                                          |                               | <p>Parameters:</p> <ul style="list-style-type: none"> <li>sXML: String - the XML of the defined searches; you can get this XML by performing an <i>export</i> of the searches from the Manage Searches dialog in Enterprise Architect</li> </ul>                                                                                                                                                                                                                                                                                                                                                                          |                                         |
| <b>AddPerspective (string Perspective, long Options)</b> | <i>Boolean</i>                | <b>Deprecated</b> - no longer in use.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                         |
| <b>AddTab (string TabName, string ControlID)</b>         | <i>activeX custom control</i> | <p>Adds an ActiveX custom control as a tabbed window. Enterprise Architect creates a control and, if successful, returns its Unknown pointer, which can be used by the caller to manipulate the control.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>TabName: String - used as the tab caption</li> <li>ControlID: String - the ProgID of the control; for example, Project1, UserControl1</li> </ul>                                                                                                                                                                                                   |                                         |
| <b>AddWindow (string WindowName, string ControlID)</b>   | <i>activeX custom control</i> | <p>Adds an ActiveX custom control as a window to the Add-Ins docked window. Enterprise Architect creates a control and, if successful, returns its Unknown pointer, which can be used by the caller to manipulate the control.</p> <p>The window can be shown by selecting it from the list in the Workspace Layouts toolbar - click on the third icon from the right and look at the end of the list.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>WindowName: String - used as the window title</li> <li>ControlID: String - the ProgID of the control; for example, Project1, UserControl1</li> </ul> | <a href="#">Workspace Layouts</a> [149] |
| <b>AdviseConnectorChange (long ConnectorID)</b>          |                               | <p>Provides an Add-In or automation client with the ability to advise the Enterprise Architect user interface that a particular connector has changed and, if it is visible in any open diagram, to reload and refresh that connector for the user.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ConnectorID: Long - the ID of the connector</li> </ul>                                                                                                                                                                                                                                                  |                                         |

| Method                                                           | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also                                                                                                                                                                |
|------------------------------------------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AdviseElementChange (long ObjectID)</b>                       |                | <p>Provides an Add-In or automation client with the ability to advise the Enterprise Architect user interface that a particular element has changed and, if it is visible in any open diagram, to reload and refresh that element for the user.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ObjectID: Long - the ID of the element</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                         |
| <b>ChangeLoginUser (string Name, string Password)</b>            | <i>Boolean</i> | <p>Sets the currently logged on user to be that specified by a name and password; this logs the user into the repository when security is enabled.</p> <p>If security is not enabled an exception (<i>Security not enabled</i>) is thrown.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Name: String - the name of the user</li> <li>Password: String - the password of the user</li> </ul>                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                         |
| <b>ClearAuditLogs (Object StartDateTime, Object EndDateTime)</b> | <i>Boolean</i> | <p>Clears all Audit Logs from the model.</p> <p>If <i>StartDateTime</i> and <i>EndDateTime</i> are not null then only log items that fall into this period are cleared.</p> <p>Returns <b>true</b> for success, <b>false</b> for failure.</p> <ul style="list-style-type: none"> <li>This method cannot be undone. It is strongly advised that you call <i>SaveAuditLogs</i> first to backup the logs</li> <li>This method might fail if the user logged into the model does not have the correct access permission</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>StartDateTime: Variant ( DateTime ) - the earliest date and time of log entries to clear</li> <li>EndDateTime; Variant ( DateTime ) - the latest date and time of log entries to clear</li> </ul> |                                                                                                                                                                         |
| <b>ClearOutput (string Name)</b>                                 |                | <p>Removes all the text from a tab in the Output window.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Name: String - the name of the tab to remove text from</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | <p><a href="#">CreateOutputTab</a><sup>[2455]</sup></p> <p><a href="#">EnsureOutputVisible</a><sup>[2456]</sup></p> <p><a href="#">WriteOutput</a><sup>[2470]</sup></p> |
| <b>CloseAddins ()</b>                                            |                | Called by automation controllers to ensure that Add-                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                         |

| Method                                                                           | Type                                                | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See also                                                                                                                                                    |
|----------------------------------------------------------------------------------|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                  |                                                     | Ins created in .NET do not linger after all controller references to Enterprise Architect have been cleared.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                             |
| <b>CloseDiagram (long DiagramID)</b>                                             |                                                     | Closes a diagram in the current list of diagrams that Enterprise Architect has open.<br><br>Parameters: <ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to close</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                             |
| <b>CloseFile ()</b>                                                              |                                                     | Closes any open file.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                             |
| <b>CreateDocumentGenerator()</b>                                                 | <a href="#">DocumentGenerator</a> <sup>[2561]</sup> | Returns a pointer to the <i>EA.DocumentGenerator</i> interface.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                             |
| <b>CreateModel (CreateModelType CreateType, string FilePath, long ParentWnd)</b> | <i>Boolean</i>                                      | Creates a new .eap model file based on the standard Enterprise Architect Base model, or a shortcut .eap based on a provided SQL connection.<br><br>Returns <b>true</b> when the new file is created, otherwise returns <b>false</b> .<br><br>Parameters: <ul style="list-style-type: none"> <li>CreateType: CreateModelType - Specify whether to make a new copy of the EABase.eap model, or create a .eap file shortcut to a DBMS repository; the latter option requires a dialog to be opened for the user to provide SQL connection details</li> <li>FilePath: String - Destination for new .eap file</li> <li>ParentWnd: Long - Window handle to act as the parent for the SQL connection dialog; only required when using <i>cmEAPFromSQLRepository</i></li> </ul> | <a href="#">CreateModelType</a> <sup>[2416]</sup>                                                                                                           |
| <b>CreateOutputTab (string Name)</b>                                             |                                                     | Creates a tab in the Output window.<br><br>Parameters: <ul style="list-style-type: none"> <li>Name: String - the name of the tab to create</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | <a href="#">CreateOutputTab</a> <sup>[2455]</sup><br><a href="#">EnsureOutputVisible</a> <sup>[2456]</sup><br><a href="#">WriteOutput</a> <sup>[2470]</sup> |
| <b>DeletePerspective (string)</b>                                                | <i>Boolean</i>                                      | <b>Deprecated</b> - no longer in use.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                             |

| Method                                                                    | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | See also                                                                                                                                                    |
|---------------------------------------------------------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Perspective, long Options)</b>                                         |                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                             |
| <b>DeleteTechnology (string ID)</b>                                       | <i>Boolean</i> | <p>Removes a specified MDG Technology resource from the repository.</p> <p>Returns <b>true</b>, if the technology is successfully removed from the model. Returns <b>false</b> otherwise.</p> <ul style="list-style-type: none"> <li>This applies to technologies imported into pre-7.0 versions of Enterprise Architect (imported technologies), not to technologies referenced in version 7.0 and later (referenced technologies)</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ID: String - the ID of the technology</li> </ul>                    | <a href="#">Deploy an MDG Technology</a> <sup>[1357]</sup>                                                                                                  |
| <b>EnsureOutputVisible (string Name)</b>                                  |                | <p>Ensures that a specified tab in the Output window is visible to the user. The Output window is made visible if it is hidden.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Name: String - the name of the tab to make visible</li> </ul>                                                                                                                                                                                                                                                                                                                  | <a href="#">CreateOutputTab</a> <sup>[2455]</sup><br><a href="#">EnsureOutputVisible</a> <sup>[2456]</sup><br><a href="#">WriteOutput</a> <sup>[2470]</sup> |
| <b>ExecutePackageBuildScript (long ScriptOptions, string PackageGuid)</b> |                | <p>Enables you to run the active package build script based on your current selection in the Project Browser. You can also run a script by passing in the package GUID.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ScriptOptions: Long - the script type; can be any one of these numerical values: <ul style="list-style-type: none"> <li>1 = Build</li> <li>2 = Test</li> <li>3 = Run</li> <li>4 = Create Workbench Instance</li> <li>5 = Debug</li> </ul> </li> <li>PackageGuid: String - the ID of the package for which to run the script</li> </ul> |                                                                                                                                                             |
| <b>Exit</b>                                                               |                | Shuts down Enterprise Architect immediately. Used                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                             |

| Method                                     | Type                                                | Notes                                                                                                                                                                                                                                                                                                                                                                                      | See also |
|--------------------------------------------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                            |                                                     | by .NET programmers where the garbage collector does not immediately release all referenced COM objects.                                                                                                                                                                                                                                                                                   |          |
| <b>GetActivePerspective ()</b>             | <i>String</i>                                       | <b>Deprecated</b> - no longer in use.                                                                                                                                                                                                                                                                                                                                                      |          |
| <b>GetAttributeByGuid (string Guid)</b>    | <a href="#">Attribute</a><br><small>[2392]</small>  | Returns a pointer to an attribute in the repository, located by its GUID. This is usually found using the <i>AttributeGUID</i> property of an attribute.<br><br>Parameters:<br><ul style="list-style-type: none"> <li>Guid: String - the GUID of the attribute to locate</li> </ul>                                                                                                        |          |
| <b>GetAttributeByID (string Id)</b>        | <a href="#">Attribute</a><br><small>[2392]</small>  | Returns a pointer to an attribute in the repository, located by its ID. This is usually found using the <i>AttributeID</i> property of an attribute.<br><br>Parameters:<br><ul style="list-style-type: none"> <li>Id: String - the ID of the attribute to locate</li> </ul>                                                                                                                |          |
| <b>GetConnectorByGuid (string Guid)</b>    | <a href="#">Connect</a><br><small>or [2508]</small> | Returns a pointer to a connector in the repository, located by its GUID. This is usually found using the <i>ConnectorGUID</i> property of a connector.<br><br>Parameters:<br><ul style="list-style-type: none"> <li>Guid: String - the GUID of the connector to locate</li> <li></li> </ul>                                                                                                |          |
| <b>GetConnectorByID (long ConnectorID)</b> | <a href="#">Connect</a><br><small>or [2508]</small> | Searches the repository for a connector with a specific ID.<br><br>Parameters:<br><ul style="list-style-type: none"> <li>ConnectorID: Long - the ID of the connector to locate</li> </ul>                                                                                                                                                                                                  |          |
| <b>GetContextItem (object Item)</b>        | <a href="#">ObjectType</a><br><small>[2419]</small> | Sets a pointer to an item in context within Enterprise Architect.<br><br>Also returns the corresponding <i>ObjectType</i> .<br><br>For additional information about <i>ContextItems</i> and the supported <i>ObjectTypes</i> see the <b>GetContextItemType</b> method (below).<br><br>Parameters:<br><ul style="list-style-type: none"> <li>Item: Object - the item to point to</li> </ul> |          |

| Method                                                        | Type                                              | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also |
|---------------------------------------------------------------|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>GetCurrentItem<br/>Type ()</b>                             | <a href="#">ObjectTy<br/>pe</a> <sup>[2419]</sup> | <p>Returns the <i>ObjectType</i> of an item in context within Enterprise Architect. A <i>ContextItem</i> is defined as an item selected anywhere within the Enterprise Architect GUI including:</p> <ul style="list-style-type: none"> <li>• An item selected in the Project Browser</li> <li>• An item selected in an open diagram</li> <li>• An item selected in certain dialogs, such as the attribute Properties dialog</li> </ul> <p>The supported <i>ObjectTypes</i> can be any one of the following values:</p> <ul style="list-style-type: none"> <li>• <i>otElement</i></li> <li>• <i>otPackage</i></li> <li>• <i>otDiagram</i></li> <li>• <i>otAttribute</i></li> <li>• <i>otMethod</i></li> <li>• <i>otConnector</i></li> </ul> |          |
| <b>GetCurrentObject<br/>()</b>                                | <i>Object</i>                                     | Returns the current context Object.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |          |
| <b>GetCurrentCounts<br/>()</b>                                | <i>String</i>                                     | Returns a set of counts from a number of tables within the base Enterprise Architect repository. These can be used to determine whether records have been added or deleted from the tables for which information is retrieved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |          |
| <b>GetCurrentDiagram<br/>()</b>                               | <a href="#">Diagram</a> <sup>[2523]</sup>         | Returns a selected diagram.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |          |
| <b>GetCurrentLoginUs<br/>er (boolean<br/>GetGuid = false)</b> | <i>String</i>                                     | <p>If security is not enabled in the repository, an error is generated.</p> <p>If <i>GetGuid</i> is <b>True</b>, a GUID generated by Enterprise Architect representing the user is returned; otherwise the text as entered in <i>System Users/User Details/Login</i> is returned.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                      |          |
| <b>GetCurrentDiagramBy<br/>Guid (string Guid)</b>             | <a href="#">Diagram</a> <sup>[2523]</sup>         | <p>Returns a pointer to a diagram using the global reference ID (global ID). This is usually found using the diagram <i>GUID</i> property of an element, and stored for later use to open an diagram without using the collection <i>GetAt()</i> function.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Guid: String - the GUID of the diagram to</li> </ul>                                                                                                                                                                                                                                                                                                                                                            |          |

| Method                                                             | Type                                                                  | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also |
|--------------------------------------------------------------------|-----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                                    |                                                                       | locate                                                                                                                                                                                                                                                                                                                                                                                                                                                             |          |
| <b>GetDiagramByID</b><br>(long DiagramID)                          | <a href="#">Diagram</a><br><small>[2523]</small>                      | Gets a pointer to a diagram using an absolute reference number (local ID). This is usually found using the <i>DiagramID</i> property of an element, and stored for later use to open a diagram without using the collection <i>GetAt()</i> function.<br><br>Parameters: <ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to locate</li> </ul>                                                                                        |          |
| <b>GetElementByGuid</b><br>(string Guid)                           | <a href="#">Element</a><br><small>[2478]</small>                      | Returns a pointer to an element in the repository, using the element's GUID reference number (global ID). This is usually found using the <i>ElementGUID</i> property of an element, and stored for later use to open an element without using the collection <i>GetAt()</i> function.<br><br>Parameters: <ul style="list-style-type: none"> <li>Guid: String - the GUID of the element to locate</li> </ul>                                                       |          |
| <b>GetElementByID</b><br>(long ElementID)                          | <a href="#">Element</a><br><small>[2478]</small>                      | Gets a pointer to an element using an absolute reference number (local ID). This is usually found using the <i>ElementID</i> property of an element, and stored for later use to open an element without using the collection <i>GetAt()</i> function.<br><br>Parameters: <ul style="list-style-type: none"> <li>ElementID: Long - the ID of the element to locate</li> </ul>                                                                                      |          |
| <b>GetElementsByQuery</b><br>(string QueryName, string SearchTerm) | <a href="#">Collection</a><br><small>[2427]</small> (of type Element) | Enables the user to run a search in Enterprise Architect, returning the result as a collection.<br><br>For example <i>GetElementsByQuery</i> ('Simple', 'Class1'), where results contain elements with <i>Class1</i> in the <b>Name</b> and <b>Notes</b> fields.<br><br>Parameters: <ul style="list-style-type: none"> <li>QueryName: String - the name of the search to run, for example 'Simple'</li> <li>SearchTerm: String - the term to search for</li> </ul> |          |
| <b>GetElementSet</b><br>(string IDList, long Options)              | <a href="#">Collection</a><br><small>[2427]</small>                   | Returns a set of elements as a collection based on a comma-separated list of <i>ElementID</i> values. By default, if no values are provided in the IDList                                                                                                                                                                                                                                                                                                          |          |



| Method                                                 | Type          | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See also |
|--------------------------------------------------------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                        |               | <p>parameter, all objects for the entire project are returned.</p> <p>Parameters</p> <ul style="list-style-type: none"> <li>• IDList: String - a comma-separated list of <i>ElementID</i> values</li> <li>• Options: Long - modifies default behaviour of this method <ul style="list-style-type: none"> <li>• 1 - Returns empty collection when empty IDList parameter is given</li> <li>• 2 - Use IDList string as an SQL query to populate this collection</li> </ul> </li> </ul>                |          |
| <b>GetFieldFromFormat (string Format, string Text)</b> | <i>String</i> | <p>Converts a field from your preferred format to Enterprise Architect's internal format; returns the field in Enterprise Architect's internal format.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Format: String - The format to convert the field from; valid formats are: <ul style="list-style-type: none"> <li>• HTML - Full HTML</li> <li>• RTF - Rich Text Format</li> <li>• TXT - Plain text</li> </ul> </li> <li>• Text: String - The field to be converted</li> </ul> |          |
| <b>GetFormatFromField (string Format, string Text)</b> | <i>String</i> | <p>After accessing a field that contains formatting, use this method to convert it to your preferred format; returns the field in the format specified.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Format: String - The format to convert the field to; valid formats are: <ul style="list-style-type: none"> <li>• HTML - Full HTML</li> <li>• RTF - Rich Text Format</li> <li>• TXT - Plain text</li> </ul> </li> <li>• Text: String - The field to be converted</li> </ul>  |          |
| <b>GetLastError ()</b>                                 | <i>String</i> | <p>Returns a string value describing the most recent error that occurred in relation to this object.</p> <p>This function is rarely used as an exception is thrown</p>                                                                                                                                                                                                                                                                                                                              |          |

| Method                                 | Type                           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also                                     |
|----------------------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
|                                        |                                | when an error occurs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                              |
| <b>GetMailInterface()</b>              | <a href="#">MailInterface</a>  | Returns an instance of the <i>EA.MailInterface</i> ; use this interface to automate the process of creating and sending model mail messages.                                                                                                                                                                                                                                                                                                                                                                                                       | <a href="#">MailInterface Package</a> [2566] |
| <b>GetMethodByGuid (string Guid)</b>   | <a href="#">Method</a> [2400]  | Returns a pointer to a method in the repository; this is usually found using the <i>MethodGUID</i> property of a method.<br><br>Parameters: <ul style="list-style-type: none"> <li>Guid: String - the GUID of the method to look for</li> </ul>                                                                                                                                                                                                                                                                                                    |                                              |
| <b>GetMethodById (string Id)</b>       | <a href="#">Method</a> [2400]  | Returns a pointer to a method in the repository; this is usually found using the <i>MethodID</i> property of a method.<br><br>Parameters: <ul style="list-style-type: none"> <li>Id: String - the ID of the method to look for</li> </ul>                                                                                                                                                                                                                                                                                                          |                                              |
| <b>GetPackageByGuid (string Guid)</b>  | <a href="#">Package</a> [2433] | Returns a pointer to a package in the repository using the package's GUID reference number (global ID). This is usually found using the <i>PackageGUID</i> property of the package.<br><br>Each package in the model also has an associated element with the same GUID, so if you have an element with <i>Type="Package"</i> then you can load the package by calling:<br><br><i>GetPackageByGuid(Element.ElementGUID)</i><br><br>Parameters: <ul style="list-style-type: none"> <li>Guid: String - the GUID of the package to look for</li> </ul> |                                              |
| <b>GetPackageById (long PackageID)</b> | <a href="#">Package</a> [2433] | Get a pointer to a package using an absolute reference number (local ID). This is usually found using the <i>PackageID</i> property of an package, and stored for later use to open a package without using the collection <i>GetAt()</i> function.<br><br>Parameters: <ul style="list-style-type: none"> <li>PackageID: Long - the ID of the package to locate</li> </ul>                                                                                                                                                                         |                                              |
| <b>GetProjectInterface ()</b>          | <a href="#">Project</a> [2535] | Return a pointer to the <i>EA.Project interface</i> (the XML-based automation server for Enterprise                                                                                                                                                                                                                                                                                                                                                                                                                                                | <a href="#">EA.Project interface</a> [2534]  |

| Method                                           | Type                                             | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also |
|--------------------------------------------------|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                  |                                                  | Architect). Use this interface to work with Enterprise Architect using XML, and also to access utility functions for loading diagrams, running reports and so on.                                                                                                                                                                                                                                                                                                                                          |          |
| <b>GetReferenceList (string Type)</b>            | <a href="#">Reference</a> <small>[2447]</small>  | <p>Uses the list type to get a pointer to a <i>Reference List</i> object.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Type: String - specifies the list type to get; valid list types are:</li> </ul> <p><i>Diagram</i></p> <p><i>Element</i></p> <p><i>Constraint</i></p> <p><i>Requirement</i></p> <p><i>Connector</i></p> <p><i>Status</i></p> <p><i>Cardinality</i></p> <p><i>Effort</i></p> <p><i>Metric</i></p> <p><i>Scenario</i></p> <p><i>Status and</i></p> <p><i>Test</i></p> |          |
| <b>GetTechnologyVersion (string ID)</b>          | String                                           | <p>Returns the version of a specified MDG Technology resource.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ID: String - the specified technology ID</li> </ul>                                                                                                                                                                                                                                                                                                                           |          |
| <b>GetTreeSelectedElements()</b>                 | <a href="#">Collection</a> <small>[2427]</small> | Returns the set of elements currently selected in the Project Browser as a collection.                                                                                                                                                                                                                                                                                                                                                                                                                     |          |
| <b>GetTreeSelectedItem (object SelectedItem)</b> | <a href="#">ObjectType</a> <small>[2413]</small> | <p>Gets an object variable and type corresponding to the currently selected item in the tree view.</p> <p>To use this function, create a generic object variable and pass this as the parameter. Depending on the return type, cast it to a more specific type.</p> <p>The object passed back through the parameter can be a package, element, diagram, attribute or operation object.</p> <p>Parameters:</p>                                                                                              |          |

| Method                                                                       | Type                                         | Notes                                                                                                                                                                                                                                                                                                                                                                                                               | See also                                                   |
|------------------------------------------------------------------------------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
|                                                                              |                                              | <ul style="list-style-type: none"> <li>SelectedItem: Object - the object to get the variable and type for</li> </ul>                                                                                                                                                                                                                                                                                                |                                                            |
| <b>GetTreeSelectedItemType ()</b>                                            | <a href="#">ObjectType</a> <sup>[2419]</sup> | <p>Returns the type of the object currently selected in the tree. One of:</p> <ul style="list-style-type: none"> <li><i>otDiagram</i></li> <li><i>otElement</i></li> <li><i>otPackage</i></li> <li><i>otAttribute</i></li> <li><i>otMethod</i></li> </ul>                                                                                                                                                           |                                                            |
| <b>GetTreeSelectedItem ()</b>                                                | <i>Object</i>                                | The related method <b>GetTreeSelectedItem()</b> has an output parameter that is inaccessible by some scripting languages. As an alternative, this method provides the selected item through the return value.                                                                                                                                                                                                       | <a href="#">GetTreeSelectedItem</a> <sup>[2462]</sup>      |
| <b>GetTreeSelectedItemPackage ()</b>                                         | <a href="#">Package</a> <sup>[2433]</sup>    | Returns the package in which the currently selected tree view object is contained.                                                                                                                                                                                                                                                                                                                                  |                                                            |
| <b>HasPerspective (string Perspective)</b>                                   | <i>String</i>                                | <b>Deprecated</b> - no longer in use.                                                                                                                                                                                                                                                                                                                                                                               |                                                            |
| <b>ImportPackageBuildScripts (string PackageGuid, string BuildScriptXML)</b> |                                              | <p>Imports build scripts into a package in Enterprise Architect.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGuid: String - the GUID of the package into which to import the build scripts.</li> <li>BuildScriptXML: String - the build script XML data, which you can export from within Enterprise Architect</li> </ul>                                                                  |                                                            |
| <b>ImportTechnology (string Technology)</b>                                  | <i>Boolean</i>                               | <p>Installs a given MDG Technology resource into the repository.</p> <p>Returns <b>True</b>, if the technology is successfully loaded into the model. Otherwise returns <b>False</b>.</p> <p>This applies to technologies imported into pre-7.0 versions of Enterprise Architect (imported technologies), not to technologies referenced in version 7.0 and later (referenced technologies).</p> <p>Parameters:</p> | <a href="#">Deploy an MDG Technology</a> <sup>[1357]</sup> |

| Method                                                                               | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also                                              |
|--------------------------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
|                                                                                      |                | <ul style="list-style-type: none"> <li>Technology: String - the contents of the technology resource file</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                       |
| <b>InvokeFileDialog</b><br>(string FilterString,<br>long Filterindex,<br>long Flags) | <i>String</i>  | Opens the Standard Open Dialog and returns a string on success.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                       |
| <b>IsTabOpen</b> (string TabName)                                                    | <i>String</i>  | <p>Checks whether a named Enterprise Architect tabbed view is open and active. This includes open diagram windows or custom controls added using <b>Repository.AddTab()</b>.</p> <p>Returns:</p> <ul style="list-style-type: none"> <li><b>2</b> to indicate that a tab is open and active (top-most)</li> <li><b>1</b> to indicate that it is open but not top-most, or</li> <li><b>0</b> to indicate that it is not visible at all</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>TabName: String - the name of the tab to check for; TabName is case sensitive</li> </ul> | <a href="#">Repository.AddTab()</a> <sup>[2453]</sup> |
| <b>IsTechnologyEnabled</b> (string ID)                                               | <i>Boolean</i> | <p>Checks whether a specified technology is enabled in Enterprise Architect.</p> <p>Returns <b>True</b> if the MDG Technology resource is enabled. Otherwise returns <b>False</b>.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ID: String - the technology ID to check for</li> </ul>                                                                                                                                                                                                                                                                                            |                                                       |
| <b>IsTechnologyLoaded</b> (string ID)                                                | <i>Boolean</i> | <p>Checks whether a specified technology is loaded into the repository.</p> <p>Returns <b>True</b> if the MDG Technology resource is loaded into the repository. Otherwise returns <b>False</b>.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ID: String - the technology ID to check for</li> </ul>                                                                                                                                                                                                                                                                              |                                                       |
| <b>OpenDiagram</b> (long DiagramID)                                                  |                | Provides a method for an automation client or Add-In to open a diagram. The diagram is added to the tabbed list of open diagrams in the main Enterprise Architect view.                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                       |

| Method                                                               | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also                                      |
|----------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
|                                                                      |                | Parameters: <ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to open</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                               |
| <b>OpenFile (string Filename)</b>                                    | <i>Boolean</i> | This is the main point for opening an Enterprise Architect project file from an automation client, and working with the contained objects.<br><br>If the required project is a DBMS repository, and you have created a shortcut .EAP file containing the database connection string, you can call this shortcut file to access the DBMS repository.<br><br>You can also connect to a SQL database by passing in the connection string itself instead of a filename. A valid connection string can be obtained from the <b>Open Project</b> dialog by selecting a recently opened SQL repository.<br><br>Parameters: <ul style="list-style-type: none"> <li>Filename: String - the filename of the Enterprise Architect project to open.</li> </ul> | <a href="#">Open a Project</a> <sup>18†</sup> |
| <b>OpenFile2 (string FilePath, string Username, string Password)</b> | <i>Boolean</i> | As for <i>OpenFile()</i> except this enables the specification of a password.<br><br>Parameters: <ul style="list-style-type: none"> <li>Filepath: String - the file path of the Enterprise Architect project to open.</li> <li>Username: String - the user login ID</li> <li>Password: String - the user password</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                       |                                               |
| <b>RefreshModelView (long PackageID)</b>                             |                | Reloads a package or the entire model, updating the user interface.<br><br>Parameters: <ul style="list-style-type: none"> <li>PackageID: Long - the ID of the package to reload: if <b>0</b>, the entire model is reloaded; if a valid package ID, only that package is reloaded</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                               |
| <b>RefreshOpenDiagrams (boolean FullReload)</b>                      |                | Refreshes the diagram contents for all diagrams open in Enterprise Architect.<br><br>Parameters: <ul style="list-style-type: none"> <li>FullReload: Boolean - if <b>false</b> the displayed contents of elements and connectors are</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                               |

| Method                                                                                                   | Type | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also                                                         |
|----------------------------------------------------------------------------------------------------------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
|                                                                                                          |      | refreshed in each diagram; if <b>true</b> each of the diagrams is completely reloaded from the repository                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                  |
| <b>ReloadDiagram (long DiagramID)</b>                                                                    |      | <p>Reloads a specified diagram. This would commonly be used to refresh a visible diagram after code import/export or other batch process where the diagram requires complete refreshing.</p> <ul style="list-style-type: none"> <li>Calling this method within a call to <b>EA_OnNotifyContextItemModified</b> is not supported</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to be reloaded</li> </ul>                                                                                                                                                                        | <a href="#">EA_OnNotifyContextItemModified</a> <sup>[2614]</sup> |
| <b>RemoveOutputTab (string Name)</b>                                                                     |      | <p>Removes a specified tab from the Output window.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Name: String - the name of the tab to be removed</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                  |
| <b>RunModelSearch (string sQueryName, string sSearchTerm, string sSearchOptions, string sSearchData)</b> |      | <p>Runs a search, displaying the results in Enterprise Architect's Model Search window.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>sQueryName: String - the name of the search to run, for example <i>Simple</i></li> <li>sSearchTerm: String - the term to search for</li> <li>sSearchOptions: String - currently not being used</li> <li>sSearchData: String - enables you to supply a list of results in the form of XML, which is appended onto the result list in Enterprise Architect - see <b>XML Format</b>; this parameter is not mandatory so pass in an empty string to run the search as per normal</li> </ul> | <a href="#">XML Format</a> <sup>[2595]</sup>                     |
| <b>SaveAllDiagrams ()</b>                                                                                |      | Saves all open diagrams.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                  |

| Method                                                                                                                  | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also |
|-------------------------------------------------------------------------------------------------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>SaveAuditLogs</b><br>(string <b>FilePath</b> ,<br>object<br><b>StartDateTime</b> ,<br>object<br><b>EndDateTime</b> ) | <i>Boolean</i> | <p>Saves the Audit Logs contained within a model to a specified file.</p> <p>If <i>StartDateTime</i> and <i>EndDateTime</i> are not null then only log items that fall into this period are saved.</p> <p>Returns <b>true</b> for success, <b>false</b> for failure.</p> <ul style="list-style-type: none"> <li>This might fail if the user logged into the model does not have the correct access permission</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>FilePath: String - the file to save the Audit Logs to</li> <li>StartDateTime: Variant ( DateTime ) - the earliest date and time of log entries to save</li> <li>EndDateTime; Variant ( DateTime ) - the latest date and time of log entries to save</li> </ul>                                                                                                                                                                                                                                                                                                                  |          |
| <b>SaveDiagram</b> (long<br><b>DiagramID</b> )                                                                          |                | <p>Saves an open diagram; assumes the diagram is open in the main user interface Tab list.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to save</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |          |
| <b>ScanXMIAndRecon</b><br><b>cile()</b>                                                                                 |                | <p>Scans the package XMI files associated with each of the project's controlled packages and restores any diagram objects or cross-references that are detected as missing from the project.</p> <p>This function is useful in team environments where each user maintains their own private copy of the model database (i.e. multiple private EAP files) and model updates are propagated through the use of controlled packages; it provides no benefit when the model is hosted in a single shared database that is accessed by all team members.</p> <p>Each controlled package is compared with its associated XMI file and, if the cross-reference information in the model does not match the XMI, Enterprise Architect updates the model with the information from the XMI and records the update in the Output window.</p> <p>You can roll back such updates by right-clicking on the entry in the Output window and selecting the context menu option <b>Rollback Update</b> (or <b>Rollback Selected Updates</b> if multiple entries are selected).</p> |          |



| Method                                                                         | Type | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | See also |
|--------------------------------------------------------------------------------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                                                |      | <p>Closing the model clears the entries in the Output window; an entry in the Output window is also cleared as and when you roll-back the update for it.</p> <p>This functionality is invoked automatically as part of the <b>Get All Latest</b> operation.</p> <p>When working in an environment that uses a Private Model deployment and your model contains a significant number of cross-package references, it is recommended that you invoke this function from time to time, following the re-importation of controlled packages - for example, after using <b>Get Latest</b> to update a number of packages, or after performing a number of package check-outs.</p> <ul style="list-style-type: none"> <li>As a general rule, avoid running this function while you have uncommitted changes in your model</li> <li>Generally, you should: <ul style="list-style-type: none"> <li>Check-out a number of packages</li> <li>Invoke <b>ScanXMIAndReconcile</b></li> <li>Make your modifications</li> <li>Commit any outstanding changes before you check-out more packages and run <b>ScanXMIAndReconcile</b> again</li> </ul> </li> </ul> |          |
| <b>ShowDynamicHelp</b><br>(string Topic)                                       |      | <p>Shows a help topic as a view.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Topic: String - specifies the help topic</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |          |
| <b>ShowInProjectView</b><br>(object Item)                                      |      | <p>Selects a specified object in the Project Browser.</p> <p>Accepted object types are <i>Package</i>, <i>Element</i>, <i>Diagram</i>, <i>Attribute</i>, and <i>Method</i>; an exception is thrown if the object is of an invalid type.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Item: Object - the object to highlight</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |          |
| <b>ShowProfileToolbox</b><br>(string Technology, string Profile, boolean Show) |      | <p>Shows/hides the contents of a specified technology or profile in the Toolbox.</p> <p>To show/hide a profile in the Toolbox, specify the profile's ID value in the <i>Profile</i> parameter and set the <i>Technology</i> parameter to a null string.</p> <p>To show/hide a technology in the Toolbox, specify the technology's ID in the <i>Technology</i> parameter and set the <i>Profile</i> parameter to a null string.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |          |

| Method                                                          | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                       | See also                                                                                 |
|-----------------------------------------------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
|                                                                 |                | Parameters: <ul style="list-style-type: none"> <li>• Technology: String - the ID of the technology</li> <li>• Profile: String - the ID of the profile</li> <li>• Show: Boolean - if <b>true</b>, show the technology or profile; if <b>false</b>, hide the technology or profile</li> </ul>                                                                                                                 |                                                                                          |
| <b>ShowWindow (long Show)</b>                                   |                | Shows or hides Enterprise Architect.<br>Parameters: <ul style="list-style-type: none"> <li>• Show: Long</li> </ul>                                                                                                                                                                                                                                                                                          |                                                                                          |
| <b>SQLQuery (string SQL)</b>                                    | <i>String</i>  | Enables execution of a SQL <i>select</i> statement against the current repository.<br>Returns an XML formatted string value of the resulting recordset.<br>Parameters: <ul style="list-style-type: none"> <li>• SQL: String - contains the SQL Select statement</li> </ul>                                                                                                                                  |                                                                                          |
| <b>SynchProfile (string Profile, string Stereotype)</b>         | <i>Boolean</i> | Synchronizes Tagged Values and constraints of a UML Profile item using the Synch Profile Elements dialog.<br>Parameters: <ul style="list-style-type: none"> <li>• Profile: String - the name of the profile that contains the stereotype</li> <li>• Stereotype: String - the name of the profile stereotype for which the default tags and constraints are to be synchronized</li> </ul>                    | <a href="#">Synchronize Tagged Values and Constraints</a> <sup>[128]</sup>               |
| <b>VersionControlResynchPkgStatuses (boolean ClearSettings)</b> |                | Synchronizes the version control status of each version controlled package within the current model with the status reported by your version control provider.<br>Parameters: <ul style="list-style-type: none"> <li>• ClearSettings: Boolean</li> </ul> <p>- if <b>true</b>, clear the version control settings from packages that are reported by the version control provider as <i>uncontrolled</i></p> | <a href="#">Resynchronize the Status of Version Controlled Packages</a> <sup>[380]</sup> |

| Method                                                   | Type | Notes                                                                                                                                                                                                                                                                                                                                             | See also                                                                                                                                                     |
|----------------------------------------------------------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                          |      | - if <b>false</b> , leave the version control settings unchanged for packages reported as <i>uncontrolled</i>                                                                                                                                                                                                                                     |                                                                                                                                                              |
| <b>WriteOutput (string Name, string String, long ID)</b> |      | Writes text to a specified tab in the Output window, and associates the text with an ID.<br>Parameters: <ul style="list-style-type: none"> <li>Name: String - specifies the tab on which to display the text.</li> <li>String: String - specifies the text to display</li> <li>ID: Long - specifies the ID the text is associated with</li> </ul> | <a href="#">ClearOutput</a> <sup>[2454]</sup><br><a href="#">CreateOutputTab</a> <sup>[2455]</sup><br><a href="#">EnsureOutput Visible</a> <sup>[2456]</sup> |

#### 19.2.2.4.15 Stereotype Class

The *Stereotype* element corresponds to a UML stereotype, which is an extension mechanism for varying the behavior and type of a model element. Use the Repository *Stereotypes* collection to add new elements and delete existing ones.

**Associated table in .EAP file:** *t\_stereotypes*

#### Stereotype Attributes:

| Attribute               | Type          | Notes                                                                                                                                                                                                          |
|-------------------------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AppliesTo</b>        | <i>String</i> | Read/Write<br>A reference to the stereotype <i>Base Class</i> , that is, which element it applies to.                                                                                                          |
| <b>MetafileLoadPath</b> | <i>String</i> | Read/Write<br>The path to an associated metafile. The automation interface does not yet support loading metafiles. To do this you must use the Stereotype tab of the UML Types dialog in Enterprise Architect. |
| <b>Notes</b>            | <i>String</i> | Read/Write.<br>Notes about the stereotype.                                                                                                                                                                     |
| <b>Name</b>             | <i>String</i> | Read/Write<br>The stereotype name, which appears in the <b>Stereotype</b> drop list for elements that match the <i>AppliesTo</i> attribute.                                                                    |

| Attribute             | Type                                 | Notes                                                                                                              |
|-----------------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| <b>ObjectType</b>     | <a href="#">ObjectType</a><br>[2419] | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                        |
| <b>StereotypeGUID</b> | <i>String</i>                        | Read/Write<br>A unique identifier for stereotype, generally set and maintained by Enterprise Architect.            |
| <b>Style</b>          | <i>String</i>                        | Read/Write<br>An additional style specifier for the stereotype.                                                    |
| <b>VisualType</b>     | <i>String</i>                        | Read/Write<br>Indicates an inbuilt visual style associated with a stereotype.<br><b>Not currently implemented.</b> |

**Stereotype Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current stereotype object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.    |

**19.2.2.4.16 Task Class**

A Task is an entry in the System ToDo list. Tasks can be accessed using the Repository *Tasks* collection.

**Associated table in .EAP file:** *t\_tasks*

**Task Attributes:**

| Attribute         | Type                                                | Notes                                                                                 |
|-------------------|-----------------------------------------------------|---------------------------------------------------------------------------------------|
| <b>ActualTime</b> | <i>Long</i>                                         | Read/Write<br>The time already expended on the task, in hours, days or other units.   |
| <b>AssignedTo</b> | <i>String</i>                                       | Read/Write<br>The person this task is assigned to; that is, the responsible resource. |
| <b>EndDate</b>    | <i>Date</i>                                         | Read/Write<br>The date the task is scheduled to finish.                               |
| <b>History</b>    | <i>String</i>                                       | Read/Write<br>A memo field to hold, for example, task history or notes.               |
| <b>Name</b>       | <i>Variant</i>                                      | Read/Write<br>The task name.                                                          |
| <b>Notes</b>      | <i>Variant</i>                                      | Read/Write<br>A description of the task.                                              |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.           |
| <b>Owner</b>      | <i>String</i>                                       | Read/Write<br>The task owner.                                                         |
| <b>Percent</b>    | <i>Long</i>                                         | Read/Write<br>The percentage completion of the task.                                  |
| <b>Phase</b>      | <i>String</i>                                       | Read/Write<br>The phase of the project the task relates to.                           |
| <b>Priority</b>   | <i>String</i>                                       | Read/Write<br>The priority of this task.                                              |
| <b>StartDate</b>  | <i>Date</i>                                         | Read/Write<br>The date the task is to start.                                          |

| Attribute        | Type           | Notes                                                                                        |
|------------------|----------------|----------------------------------------------------------------------------------------------|
| <b>Status</b>    | <i>Variant</i> | Read/Write<br>The current status of the task.                                                |
| <b>TaskID</b>    | <i>Long</i>    | Read only<br>The local ID of the task.                                                       |
| <b>TotalTime</b> | <i>Long</i>    | Read/Write<br>The total expected time the task might run, in hours, days or some other unit. |
| <b>Type</b>      | <i>String</i>  | Read/Write<br>Sets or returns a string representing the type.                                |

**Task Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current Task object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.          |

**19.2.2.4.17 Term Class**

A *Term* object represents one entry in the system glossary. Terms can be accessed using the Repository *Terms* collection.

**Associated table in .EAP file:** *t\_glossary*

**Term Attributes:**

| Attribute      | Type          | Notes                                                   |
|----------------|---------------|---------------------------------------------------------|
| <b>Meaning</b> | <i>String</i> | Read/Write<br>The description of the term; its meaning. |

| Attribute         | Type                                 | Notes                                                                             |
|-------------------|--------------------------------------|-----------------------------------------------------------------------------------|
| <b>ObjectType</b> | <a href="#">ObjectType</a><br>[2419] | Read only<br>Distinguishes objects referenced through a Dispatch interface.       |
| <b>Term</b>       | <i>String</i>                        | Read/Write<br>The glossary item name.                                             |
| <b>TermID</b>     | <i>Long</i>                          | Read only<br>A local ID number to identify the term in the model.                 |
| <b>Type</b>       | <i>String</i>                        | Read/Write<br>The type this term applies to (for example, business or technical). |

**Term Methods:**

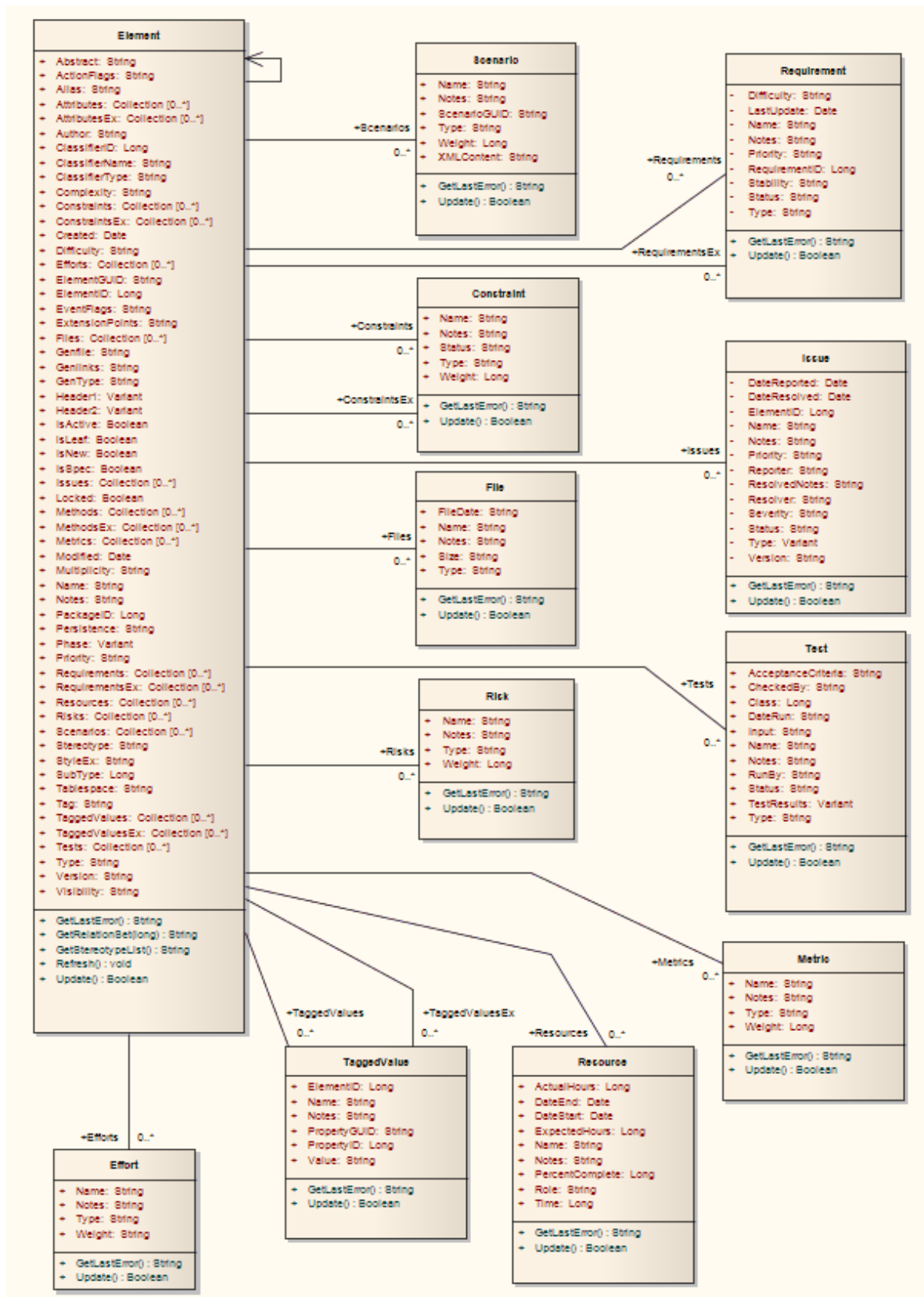
| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current Term object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.          |

**19.2.2.5 Element Package**

The *Element* package contains information about an element and its associated extended properties such as testing and project management information. An element is the basic item in an Enterprise Architect model. Classes, Use Cases and Components are all different types of UML element.

The diagram below illustrates the relationships between an *element* and its associated extended information. The related information is accessed through the collections owned by the element (for example, Scenarios and Tests). It also includes a full description of the element object (the basic model structural unit).

**Example**





### 19.2.2.5.1 Constraint Class

A *Constraint* is a condition imposed on an element. Constraints are accessed through the Element *Constraints* collection.

**Associated table in .EAP file:** *t\_objectconstraints*

#### Constraint Attributes:

| Attribute         | Type                                 | Notes                                                                              |
|-------------------|--------------------------------------|------------------------------------------------------------------------------------|
| <b>Name</b>       | <i>String</i>                        | Read/Write<br>The name of the constraint (that is, the constraint).                |
| <b>Notes</b>      | <i>String</i>                        | Read/Write<br>Notes about the constraint.                                          |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br>[2419] | Read only<br>Distinguishes objects referenced through a Dispatch interface.        |
| <b>ParentID</b>   | <i>Long</i>                          | Read only<br>The <i>ElementID</i> of the element to which this constraint applies. |
| <b>Status</b>     | <i>String</i>                        | Read/Write<br>The current status of the constraint.                                |
| <b>Type</b>       | <i>String</i>                        | Read/Write<br>The constraint type.                                                 |
| <b>Weight</b>     | <i>Long</i>                          | Read/Write<br>A weighting factor.                                                  |

#### Constraint Methods:

| Method                 | Type          | Notes                                                                                                                                                                                 |
|------------------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i> | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |

| Method           | Type           | Notes                                                                                                                                                                                    |
|------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Update ()</b> | <i>Boolean</i> | Update the current <i>Constraint</i> object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

#### 19.2.2.5.2 Effort Class

An *Effort* is a named item with a weighting that can be associated with an element for purposes of building metrics about the model. Efforts are accessed through the Element *Efforts* collection.

Associated table in .EAP file: *t\_objecteffort*

#### Effort Attributes:

| Attribute         | Type                                                | Notes                                                                       |
|-------------------|-----------------------------------------------------|-----------------------------------------------------------------------------|
| <b>Name</b>       | <i>String</i>                                       | Read/Write<br>The name of the effort.                                       |
| <b>Notes</b>      | <i>String</i>                                       | Read/Write<br>Notes about the effort.                                       |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface. |
| <b>Type</b>       | <i>String</i>                                       | Read/Write<br>The effort type.                                              |
| <b>Weight</b>     | <i>Long</i>                                         | Read/Write<br>A weighting factor.                                           |
| <b>Weight2</b>    | <i>Float</i>                                        | Read/Write<br>A weighting factor.                                           |

#### Effort Methods:

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Saves the effort to the model.                                                                                                                                                        |

### 19.2.2.5.3 Element Class

An *Element* is the main modeling unit. It corresponds to (for example) a Class, Use Case, Node or Component. You create new elements by adding to the Package *Elements* collection. Once you have created an element, you can add it to the *DiagramObjectClass* of a diagram to include it in the diagram.

Elements also have a collection of connectors. Each entry in this collection indicates a relationship to another element.

There are also some extended collections for managing additional information about the element, including things such as Tagged Values, Issues, Constraints and Requirements.

**Associated table in .EAP file:** *t\_object*

#### Element Attributes

| Attribute                          | Type                                         | Notes                                                                                                                                                         | See also                                                                                                                     |
|------------------------------------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| <b>Abstract</b>                    | <i>String</i>                                | Read/Write<br><br>Indicates if the element is Abstract ( <b>1</b> ) or Concrete ( <b>0</b> ).                                                                 |                                                                                                                              |
| <b>ActionFlags</b>                 | <i>String</i>                                | Read/Write<br><br>A structure to hold flags concerned with Action semantics.                                                                                  |                                                                                                                              |
| <b>Alias</b>                       | <i>String</i>                                | Read/Write<br><br>An optional alias for this element.                                                                                                         |                                                                                                                              |
| <b>AssociationClassConnectorID</b> | <i>Long</i>                                  | Read only<br><br>If the element is an AssociationClass, <i>AssociationClassConnectorID</i> contains the Connector ID of the respective Association connector. | <a href="#">CreateAssociationClass()</a> <sup>[2489]</sup><br><br>(Information on creating AssociationClasses from elements) |
| <b>Attributes</b>                  | <a href="#">Collection</a> <sup>[2427]</sup> | Read only                                                                                                                                                     |                                                                                                                              |

| Attribute               | Type                                 | Notes                                                                                                                                                                                                  | See also                                      |
|-------------------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
|                         |                                      | A collection of Attribute objects for the current element; use the <b>AddNew</b> and <b>Delete</b> functions to manage attributes.                                                                     |                                               |
| <b>AttributesEx</b>     | <a href="#">Collection</a><br>[2427] | Read only<br>A collection of Attribute objects belonging to the current element and its parent elements.                                                                                               |                                               |
| <b>Author</b>           | <i>String</i>                        | Read/Write<br>The element author.                                                                                                                                                                      | <a href="#">Repository: Authors</a><br>[2448] |
| <b>BaseClasses</b>      | <a href="#">Collection</a><br>[2427] | Read only<br>List of Base Classes for this element, presented as a collection for convenience.                                                                                                         |                                               |
| <b>ClassifierID</b>     | <i>Long</i>                          | <b>Deprecated</b><br>See <i>ClassifierID</i> .                                                                                                                                                         |                                               |
| <b>ClassifierID</b>     | <i>Long</i>                          | Read/Write<br>The ElementID of a Classifier associated with this element; that is, the base type.<br>Only valid for instance type elements (such as Object or Sequence).                               |                                               |
| <b>ClassifierName</b>   | <i>String</i>                        | Read/Write<br>Name of associated Classifier (if any).                                                                                                                                                  |                                               |
| <b>ClassifierType</b>   | <i>String</i>                        | Read only<br>Type of associated Classifier.                                                                                                                                                            |                                               |
| <b>Complexity</b>       | <i>String</i>                        | Read/Write<br>A complexity value indicating how complex the element is; used for metric reporting and estimation.<br>Valid values are: <b>1</b> for Easy, <b>2</b> for Medium, <b>3</b> for Difficult. |                                               |
| <b>CompositeDiagram</b> | <a href="#">Diagram</a><br>[2523]    | Read only<br>If the element is Composite, returns its                                                                                                                                                  |                                               |

| Attribute               | Type                                 | Notes                                                                                                                                                                                                                                                               | See also                                |
|-------------------------|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
|                         |                                      | associated diagram; otherwise returns null.                                                                                                                                                                                                                         |                                         |
| <b>Connectors</b>       | <a href="#">Collection</a><br>[2427] | Read only<br>Returns a collection containing the connectors to other elements.                                                                                                                                                                                      |                                         |
| <b>Constraints</b>      | <a href="#">Collection</a><br>[2427] | Read only<br>A collection of Constraint objects.                                                                                                                                                                                                                    | <a href="#">Constraint Class</a> [2476] |
| <b>ConstraintsEx</b>    | <a href="#">Collection</a><br>[2427] | Read only<br>Collection of Constraint objects belonging to the current element and its parent elements.                                                                                                                                                             |                                         |
| <b>Created</b>          | <i>Date</i>                          | Read/Write<br>The date the element was created.                                                                                                                                                                                                                     |                                         |
| <b>CustomProperties</b> | <a href="#">Collection</a><br>[2427] | Read only<br>List of advanced properties for an element.<br>The collection of advanced properties differs depending on element type; for example, an Action and an Activity have different advanced properties.<br>Currently only editable from the user interface. |                                         |
| <b>Diagrams</b>         | <a href="#">Collection</a><br>[2427] | Read only<br>Returns a collection of sub-diagrams (child diagrams) attached to this element as seen in the tree view.                                                                                                                                               |                                         |
| <b>Difficulty</b>       | <i>String</i>                        | Read/Write<br>A difficulty level associated with this element for estimation/metrics; only useable for Requirement, Change and Issue element types, otherwise ignored.<br>Valid values are: <b>Low</b> , <b>Medium</b> , <b>High</b> .                              |                                         |
| <b>Efforts</b>          | <a href="#">Collection</a><br>[2427] | Read only<br>A collection of Effort objects.                                                                                                                                                                                                                        | <a href="#">Effort Class</a> [2477]     |

| Attribute               | Type                                                 | Notes                                                                                                                                                                     | See also                                            |
|-------------------------|------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| <b>ElementGUID</b>      | <i>String</i>                                        | Read only<br>A globally unique ID for this element; that is, unique across all model files.                                                                               |                                                     |
| <b>ElementID</b>        | <i>Long</i>                                          | Read only<br>The local ID of the Element; valid for this file only.                                                                                                       |                                                     |
| <b>Elements</b>         | <a href="#">Collection</a><br><small>[2427]</small>  | Read only<br>Returns a collection of child elements (sub-elements) attached to this element as seen in the tree view.                                                     |                                                     |
| <b>EmbeddedElements</b> | <a href="#">Collections</a><br><small>[2427]</small> | Read only<br>A list of elements that are embedded into this element, such as Ports, Parts, Pins and Parameter Sets.                                                       |                                                     |
| <b>EventFlags</b>       | <i>String</i>                                        | Read/Write<br>A structure to hold a variety of flags to do with signals or events.                                                                                        |                                                     |
| <b>ExtensionPoints</b>  | <i>String</i>                                        | Read/Write<br>Optional extension points for a Use Case as a comma-separated list.                                                                                         |                                                     |
| <b>Files</b>            | <a href="#">Collection</a><br><small>[2427]</small>  | Read only<br>A collection of <b>File</b> objects.                                                                                                                         | <a href="#">File Class</a><br><small>[2491]</small> |
| <b>GenFile</b>          | <i>String</i>                                        | Read/Write<br>The file associated with this element for code generation and synchronization purposes; can include macro expansion tags for local conversion to full path. |                                                     |
| <b>Genlinks</b>         | <i>String</i>                                        | Read/Write<br>Links to other Classes discovered at code reversing time; Parents and Implements connectors only.                                                           |                                                     |
| <b>GenType</b>          | <i>String</i>                                        | Read/Write                                                                                                                                                                |                                                     |


| Attribute          | Type                                               | Notes                                                                                                                       | See also |
|--------------------|----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|----------|
|                    |                                                    | The code generation type; for example, Java, C++, C#, VBNet, Visual Basic, Delphi.                                          |          |
| <b>Header1</b>     | <i>Variant</i>                                     | Read/Write<br>A user defined string for inclusion as header in the source files generated.                                  |          |
| <b>Header2</b>     | <i>Variant</i>                                     | Read/Write<br>Same as for <b>Header1</b> , but used in the CPP source file.                                                 |          |
| <b>IsActive</b>    | <i>Boolean</i>                                     | Read/Write<br>Boolean value indicating whether the element is active or not.<br><b>1</b> = True, <b>0</b> = False.          |          |
| <b>IsComposite</b> | <i>Boolean</i>                                     | Read/Write<br>Indicates whether the element is composite or not.<br><b>1</b> = True, <b>0</b> = False.                      |          |
| <b>IsLeaf</b>      | <i>Boolean</i>                                     | Read/Write<br>Boolean value indicating whether the element is in leaf node or not.<br><b>1</b> = True, <b>0</b> = False.    |          |
| <b>IsNew</b>       | <i>Boolean</i>                                     | Read/Write<br>Boolean value indicating whether the element is new or not.<br><b>1</b> = True, <b>0</b> = False.             |          |
| <b>IsSpec</b>      | <i>Boolean</i>                                     | Read/Write<br>Boolean value indicating whether the element is a specification or not.<br><b>1</b> = True, <b>0</b> = False. |          |
| <b>Issues</b>      | <a href="#">Collection</a><br><small>[242]</small> | Read only<br>Collection of Issue objects.                                                                                   |          |

| Attribute           | Type                                                | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See also |
|---------------------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Locked</b>       | <i>Boolean</i>                                      | Read/Write<br>Indicates if the element has been locked against further change.                                                                                                                                                                                                                                                                                                                                                                                                        |          |
| <b>MetaType</b>     | <i>String</i>                                       | Read only<br>The element's domain-specific meta type, as defined by an applied stereotype from an MDG Technology.                                                                                                                                                                                                                                                                                                                                                                     |          |
| <b>Methods</b>      | <a href="#">Collection</a><br><small>[2427]</small> | Read only<br>Collection of Method objects for current element.                                                                                                                                                                                                                                                                                                                                                                                                                        |          |
| <b>MethodsEx</b>    | <a href="#">Collection</a><br><small>[2427]</small> | Read only<br>Collection of Method objects belonging to the current element and its parent elements.                                                                                                                                                                                                                                                                                                                                                                                   |          |
| <b>Metrics</b>      | <a href="#">Collection</a><br><small>[2427]</small> | Read only<br>Collection of Metric elements for current element.                                                                                                                                                                                                                                                                                                                                                                                                                       |          |
| <b>MiscData</b>     | <i>String</i>                                       | Read only<br>This low-level property provides information about the contents of the <b>PData<sub>x</sub></b> fields.<br>These database fields are not documented, and developers must gain understanding of these fields through their own endeavors to use this property.<br><b>MiscData</b> is zero based, therefore: <ul style="list-style-type: none"> <li>• <b>MiscData(0)</b> corresponds to <b>PData1</b></li> <li>• <b>MiscData(1)</b> to <b>PData2</b></li> </ul> and so on. |          |
| <b>Modified</b>     | <i>Date</i>                                         | Read/Write<br>The date the element was last modified.                                                                                                                                                                                                                                                                                                                                                                                                                                 |          |
| <b>Multiplicity</b> | <i>String</i>                                       | Read/Write<br>Multiplicity value for this element.                                                                                                                                                                                                                                                                                                                                                                                                                                    |          |
| <b>Name</b>         | <i>String</i>                                       | Read/Write<br>The element name; should be unique within the                                                                                                                                                                                                                                                                                                                                                                                                                           |          |



| Attribute          | Type                                                 | Notes                                                                                                                                                                                                                           | See also |
|--------------------|------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                    |                                                      | current package.                                                                                                                                                                                                                |          |
| <b>Notes</b>       | <i>String</i>                                        | Read/Write<br>Further descriptive text about the element.                                                                                                                                                                       |          |
| <b>ObjectType</b>  | <a href="#">Object Type</a><br><small>[2418]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                     |          |
| <b>PackageID</b>   | <i>Long</i>                                          | Read/Write<br>A local ID for the package containing this element.                                                                                                                                                               |          |
| <b>ParentID</b>    | <i>Long</i>                                          | Read/Write<br>If this element is a child of another, used to set or retrieve the <i>ElementID</i> of the other element; if not, returns <b>0</b> .                                                                              |          |
| <b>Partitions</b>  | <a href="#">Collection</a><br><small>[2427]</small>  | Read only<br>List of logical partitions into which an element can be divided.<br><br>Only valid for elements that support partitions, such as Activities and States.                                                            |          |
| <b>Persistence</b> | <i>String</i>                                        | Read/Write<br>The persistence associated with this element; can be <b>Persistent</b> or <b>Transient</b> .                                                                                                                      |          |
| <b>Phase</b>       | <i>String</i>                                        | Read/Write<br>The phase this element is scheduled to be constructed in; any string value.                                                                                                                                       |          |
| <b>Priority</b>    | <i>String</i>                                        | Read/Write<br>The priority of this element as compared to other project elements; only applies to Requirement, Change and Issue types, otherwise ignored.<br><br>Valid values are: <b>Low</b> , <b>Medium</b> and <b>High</b> . |          |
| <b>Properties</b>  | <a href="#">Properties</a><br><small>[2410]</small>  | Returns a list of specialized properties that apply to the element that might not be available using                                                                                                                            |          |

| Attribute               | Type                                                | Notes                                                                                                                                                                                              | See also                                                   |
|-------------------------|-----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
|                         |                                                     | <p>the automation model.</p> <p>The properties are purposely undocumented because of their obscure nature and because they are subject to change as progressive enhancements are made to them.</p> |                                                            |
| <b>PropertyType</b>     | <i>Long</i>                                         | <p>Read/Write</p> <p>The ElementID of a Type associated with this element; only valid for Port and Part elements.</p>                                                                              |                                                            |
| <b>Realizes</b>         | <a href="#">Collection</a><br><small>[2427]</small> | <p>Read only</p> <p>List of Interfaces realized by this element for convenience.</p>                                                                                                               |                                                            |
| <b>Requirements</b>     | <a href="#">Collection</a><br><small>[2427]</small> | <p>Read only</p> <p>Collection of <b>Requirement</b> objects.</p>                                                                                                                                  | <a href="#">Requirement Class</a><br><small>[2495]</small> |
| <b>RequirementsEx</b>   | <a href="#">Collection</a><br><small>[2427]</small> | <p>Read only</p> <p>Collection of <b>Requirement</b> objects belonging to the current element and its parent elements.</p>                                                                         | <a href="#">Requirement Class</a><br><small>[2495]</small> |
| <b>Resources</b>        | <a href="#">Collection</a><br><small>[2427]</small> | <p>Read only</p> <p>Collection of <b>Resource</b> objects for current element.</p>                                                                                                                 | <a href="#">Resource Class</a><br><small>[2497]</small>    |
| <b>Risks</b>            | <a href="#">Collection</a><br><small>[2427]</small> | <p>Read only</p> <p>Collection of <b>Risk</b> objects.</p>                                                                                                                                         | <a href="#">Risk Class</a><br><small>[2498]</small>        |
| <b>RunState</b>         | <i>String</i>                                       | <p>Read/Write</p> <p>The object's runstate list as a string.</p>                                                                                                                                   |                                                            |
| <b>Scenarios</b>        | <a href="#">Collection</a><br><small>[2427]</small> | <p>Read only</p> <p>Collection of <b>Scenario</b> objects for current element.</p>                                                                                                                 | <a href="#">Scenario Class</a><br><small>[2499]</small>    |
| <b>StateTransitions</b> | <a href="#">Collection</a><br><small>[2427]</small> | <p>Read only</p> <p>List of State Transitions that an element can support; applies in particular to Timing elements.</p>                                                                           |                                                            |

| Attribute           | Type          | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also                                                                                                   |
|---------------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| <b>Status</b>       | <i>String</i> | Read/Write<br>Sets or gets the status, such as <b>Proposed</b> or <b>Approved</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                            |
| <b>Stereotype</b>   | <i>String</i> | Read/Write<br>The primary element stereotype; the first of the list of stereotypes you can access using the <i>StereotypeEx</i> attribute.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                            |
| <b>StereotypeEx</b> | <i>String</i> | Read/Write<br>All the applied stereotypes of the element in a comma-separated list.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                            |
| <b>StyleEx</b>      | <i>String</i> | Read/Write<br>Advanced style settings; reserved for the use of Sparx Systems.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                            |
| <b>Subtype</b>      | <i>Long</i>   | Read/Write<br>A numeric subtype that qualifies the <b>Type</b> of the main element <ul style="list-style-type: none"> <li>• For Event: <b>0</b> = Receiver, <b>1</b> = Sender</li> <li>• For Class: <b>1</b> = Parameterised, <b>2</b> = Instantiated, <b>3</b> = Both, <b>0</b> = Neither, <b>17</b> = Association Class</li> </ul> <p>If <b>17</b>, because an Association Class has been created through the user interface, <b>MiscData(3)</b> contains the ID of the related Association; as MiscData is read-only, you cannot create an Association Class through the Automation Interface:</p> <ul style="list-style-type: none"> <li>• For Note: <b>1</b> = Note linked to connector, <b>2</b> = Constraint linked to connector</li> <li>• For StateNode: <b>100</b> = ActivityInitial, <b>101</b> = ActivityFinal</li> <li>• For Activity: <b>0</b> = Activity, <b>8</b> = composite Activity (also set to <b>8</b> for other composite elements such as Use Cases)</li> <li>• For Synchronization: <b>0</b> = Horizontal, <b>1</b> = Vertical</li> </ul> | <a href="#">Type</a>  |

| Attribute             | Type                                                                                               | Notes                                                                                                                                                                                                     | See also                                                                                                                                                                                      |
|-----------------------|----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       |                                                                                                    | Note that there are many more Types than indicated in the above examples.                                                                                                                                 |                                                                                                                                                                                               |
| <b>Tablespace</b>     | <i>String</i>                                                                                      | Read/Write<br>Associated tablespace for a Table element.                                                                                                                                                  |                                                                                                                                                                                               |
| <b>Tag</b>            | <i>String</i>                                                                                      | Read/Write<br>Corresponds to the <b>Keywords</b> field in the Enterprise Architect user interface.                                                                                                        | <a href="#">General Settings</a> <sup>[845]</sup>                                                                                                                                             |
| <b>TaggedValues</b>   | <a href="#">Collection</a> <sup>[2427]</sup> of type <a href="#">TaggedValue</a> <sup>[2504]</sup> | Read only<br>Returns a collection of TaggedValue objects.                                                                                                                                                 | <a href="#">TaggedValue Class</a> <sup>[2504]</sup>                                                                                                                                           |
| <b>TaggedValuesEx</b> | <a href="#">Collection</a> <sup>[2427]</sup> of type <a href="#">TaggedValue</a> <sup>[2504]</sup> | Read only<br>Returns a collection of TaggedValue objects belonging to the current element and the elements specialized or realized by the current element.                                                | <a href="#">TaggedValue Class</a> <sup>[2504]</sup>                                                                                                                                           |
| <b>Tests</b>          | <a href="#">Collection</a> <sup>[2427]</sup>                                                       | Read only<br>A collection of Test objects for the current element.                                                                                                                                        | <a href="#">Test Class</a> <sup>[2506]</sup>                                                                                                                                                  |
| <b>TreePos</b>        | <i>Long</i>                                                                                        | Read/Write<br>Sets or gets the tree position.                                                                                                                                                             |                                                                                                                                                                                               |
| <b>Type</b>           | <i>String</i>                                                                                      | Read/Write<br>The element type (such as Class, Component).<br>Note that Type is case sensitive inside Enterprise Architect and should be provided with an initial capital (proper case); valid types are: |                                                                                                                                                                                               |
|                       |                                                                                                    | <b>Action</b><br><b>Activity</b><br><b>ActivityPartition</b><br><b>ActivityRegion</b><br><b>Actor</b><br><b>Artifact</b><br><b>Association</b><br><b>Boundary</b><br><b>Change</b>                        | <b>InteractionOccurrence</b><br><b>InteractionState</b><br><b>Interface</b><br><b>InterruptibleActivityRegion</b><br><b>on</b><br><b>Issue</b><br><b>Node</b><br><b>Note</b><br><b>Object</b> |

| Attribute         | Type          | Notes                                                                                                                                                                                                                                                                                                                                                                                         | See also                                                                                                                                                                                                                                                                                                                                   |  |
|-------------------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
|                   |               | <b>Class</b><br><b>Collaboration</b><br><b>Component</b><br><b>Constraint</b><br><b>Decision</b><br><b>DeploymentSpecification</b><br><b>DiagramFrame</b><br><b>EmbeddedElement</b><br><b>Entity</b><br><b>EntryPoint</b><br><b>Event</b><br><b>ExceptionHandler</b><br><b>ExitPoint</b><br><b>ExpansionNode</b><br><b>ExpansionRegion</b><br><b>GUIElement</b><br><b>InteractionFragment</b> | <b>Package</b><br><b>Parameter</b><br><b>Part</b><br><b>Port</b><br><b>ProvidedInterface</b><br><b>Report</b><br><b>RequiredInterface</b><br><b>Requirement</b><br><b>Screen</b><br><b>Sequence</b><br><b>State</b><br><b>StateNode</b><br><b>Synchronization</b><br><b>Text</b><br><b>TimeLine</b><br><b>UMLDiagram</b><br><b>UseCase</b> |  |
| <b>Version</b>    | <i>String</i> | Read/Write<br>The version of the element.                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                            |  |
| <b>Visibility</b> | <i>String</i> | Read/Write<br>The Scope of this element within the current package.<br>Valid values are: <b>Public, Private, Protected</b> or <b>Package</b> .                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                            |  |

## Element Methods

| Method                                    | Type           | Notes                                                                                                                                                                                                                                                                                                                                                        | See also |
|-------------------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>ApplyGroupLock (string aGroupName)</b> | <i>Boolean</i> | Applies a group lock to the element object, for the specified group, on behalf of the current user.<br>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information.<br>Parameters: <ul style="list-style-type: none"> <li>aGroupName: String - the name of the user group for which to set the group lock</li> </ul> |          |
| <b>ApplyUserLock ()</b>                   | <i>Boolean</i> | Applies a user lock to the element object for the current user.<br>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information.                                                                                                                                                                                      |          |

| Method                                           | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See also                                                                                                      |
|--------------------------------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| <b>CreateAssociationClass (long ConnectorID)</b> | <i>Boolean</i> | <p>Makes this element an AssociationClass of the Association with the provided <b>Connector ID</b>; the return value indicates whether the function succeeded in converting the element to an AssociationClass.</p> <p>AssociationClasses are created only where:</p> <ul style="list-style-type: none"> <li>• The current element is valid</li> <li>• The current element is a Class</li> <li>• The current element is not already an AssociationClass</li> <li>• The specified connector exists</li> <li>• The specified connector is an Association</li> <li>• The specified connector is not already in an AssociationClass pair</li> <li>• The current element is not at either end of the specified connector</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• ConnectorID: Long - the Connector ID of an Association connector</li> </ul> |                                                                                                               |
| <b>GetLastError ()</b>                           | <i>String</i>  | <p>Returns a string value describing the most recent error that occurred in relation to this object.</p> <p>This function is rarely used as an exception is thrown when an error occurs.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                               |
| <b>GetLinkedDocument ()</b>                      | <i>String</i>  | <p>Returns a string value containing the element's linked document contents, in RTF format.</p> <p>If the element contains no linked document, an empty string is returned.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                               |
| <b>GetRelationSet (EnumRelationSetType Type)</b> | <i>String</i>  | <p>Returns a string containing a comma-separated list of ElementIDs of directly- and indirectly-related elements based on the given type.</p> <p>Recurses using the same relation type on all elements it finds, retrieving all dependencies and sub-dependencies of the current element; for example, <i>Object1</i> depends on <i>Object2</i>, which depends on <i>Object3</i>, therefore this method returns <i>Object2</i> <b>and</b> <i>Object3</i>.</p> <p>To obtain only the direct relationships of the</p>                                                                                                                                                                                                                                                                                                                                                    | <p><a href="#">EnumRelationSetType</a><sup>[2417]</sup></p> <p><a href="#">Connector</a><sup>[2507]</sup></p> |

| Method                                                   | Type           | Notes                                                                                                                                                                                                                                                                                                             | See also                                            |
|----------------------------------------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
|                                                          |                | element, use the <i>Connector</i> collection instead.                                                                                                                                                                                                                                                             |                                                     |
| <b>GetStereotypeList ()</b>                              | <i>String</i>  | Returns a comma-separated list of stereotypes allied to this element.                                                                                                                                                                                                                                             |                                                     |
| <b>IsAssociationClass</b>                                | <i>Boolean</i> | Returns whether or not the current element is an <i>AssociationClass</i> .                                                                                                                                                                                                                                        |                                                     |
| <b>LoadLinkedDocument (string Filename)</b>              | <i>Boolean</i> | Loads the RTF document from the specified file into the element's linked document.<br><br>Parameters: <ul style="list-style-type: none"> <li>• <b>FileName:</b> <i>String</i> - the name of the file from which to load the RTF document</li> </ul>                                                               |                                                     |
| <b>Refresh ()</b>                                        | <i>Void</i>    | Refreshes the element features in the Project Browser.<br><br>Usually called after adding or deleting attributes or methods, when the user interface is required to be updated as well.                                                                                                                           |                                                     |
| <b>ReleaseUserLock ()</b>                                | <i>Boolean</i> | Releases a user lock or group lock on the element object.<br><br>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information.                                                                                                                                             |                                                     |
| <b>SaveLinkedDocument (string Filename)</b>              | <i>Boolean</i> | Saves the linked document for this element to the specified RTF file.<br><br>Parameters: <ul style="list-style-type: none"> <li>• <b>FileName:</b> <i>String</i> - the name of the RTF file to which to save the linked document</li> </ul>                                                                       |                                                     |
| <b>SetAppearance (long Scope, long Item, long Value)</b> | <i>Void</i>    | Sets the visual appearance of the element.<br><br>Parameters: <ul style="list-style-type: none"> <li>• <b>Scope:</b> <i>Long</i> - Scope of appearance set to modify<br/>1 – Base (Default appearance across entire model)</li> </ul><br>To set appearance for the element (diagram object) in a selected diagram | <a href="#">DiagramObject Class</a> <sup>2537</sup> |

| Method                           | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                | See also                                                   |
|----------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
|                                  |                | <p>only, see <i>Setting The Style</i> in the <i>Diagram Object Class</i> topic</p> <ul style="list-style-type: none"> <li>Item: Long - Appearance feature to modify <ul style="list-style-type: none"> <li>0 – Background color</li> <li>1 – Font Color</li> <li>2 – Border Color</li> <li>3 – Border Width</li> </ul> </li> <li>Value: Long - Value to set appearance to</li> </ul> |                                                            |
| <b>SetCreated (Date NewVal)</b>  | <i>Void</i>    | <b>Deprecated</b><br>This method is no longer supported.                                                                                                                                                                                                                                                                                                                             |                                                            |
| <b>SetModified (Date NewVal)</b> | <i>Void</i>    | <b>Deprecated</b><br>This method is no longer supported.                                                                                                                                                                                                                                                                                                                             |                                                            |
| <b>UnlinkFromAssociation</b>     | <i>Boolean</i> | Performs the opposite of <b>CreateAssociationClass()</b> .                                                                                                                                                                                                                                                                                                                           | <a href="#">CreateAssociationClass()</a> <sup>[2489]</sup> |
| <b>Update ()</b>                 | <i>Boolean</i> | Updates the current element object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.                                                                                                                                                                                                      |                                                            |

#### 19.2.2.5.4 File Class

A *File* represents an associated file for an element. Files are accessed through the Element *Files* collection.

**Associated table in .EAP file:** *t\_objectfiles*

##### **File Attributes:**

| Attribute       | Type          | Notes                                                   |
|-----------------|---------------|---------------------------------------------------------|
| <b>FileDate</b> | <i>String</i> | Read/Write<br>The file date when the entry was created. |
| <b>Name</b>     | <i>String</i> | Read/Write                                              |



| Attribute         | Type                                             | Notes                                                                                        |
|-------------------|--------------------------------------------------|----------------------------------------------------------------------------------------------|
|                   |                                                  | The file name can be a logical file or a reference to a web address (using <i>http://</i> ). |
| <b>Notes</b>      | <i>String</i>                                    | Read/Write<br>Notes about the file.                                                          |
| <b>ObjectType</b> | <a href="#">ObjectTyp</a><br>e <sup>[2419]</sup> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                  |
| <b>Size</b>       | <i>String</i>                                    | Read/Write<br>The file size.                                                                 |
| <b>Type</b>       | <i>String</i>                                    | Read/Write<br>The file type.                                                                 |

**File Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current File object after modification or appending a new item.<br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.              |

**19.2.2.5.5 Issue (Maintenance) Class**

An *Issue* is either a *Change* or a *Defect*, is associated with the containing element, and is accessed through the *Issues* collection of an element.

**Associated table in .EAP file:** *t\_objectproblems*

**Issue Attributes:**

| Attribute           | Type        | Notes      |
|---------------------|-------------|------------|
| <b>DateReported</b> | <i>Date</i> | Read/Write |

| Attribute            | Type                                                | Notes                                                                                 |
|----------------------|-----------------------------------------------------|---------------------------------------------------------------------------------------|
|                      |                                                     | The date the issue was reported.                                                      |
| <b>DateResolved</b>  | <i>Date</i>                                         | Read/Write<br>The date the issue was resolved.                                        |
| <b>ElementID</b>     | <i>Long</i>                                         | Read/Write<br>The ID of the element associated with this issue.                       |
| <b>Name</b>          | <i>String</i>                                       | Read/Write<br>The Issue name; that is, the Issue itself.                              |
| <b>Notes</b>         | <i>String</i>                                       | Read/Write<br>The Issue description.                                                  |
| <b>ObjectType</b>    | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.           |
| <b>Priority</b>      | <i>String</i>                                       | Read/Write<br>The priority of the Issue - <b>Low</b> , <b>Medium</b> or <b>High</b> . |
| <b>Reporter</b>      | <i>String</i>                                       | Read/Write<br>The user ID of the person reporting the issue.                          |
| <b>Resolver</b>      | <i>String</i>                                       | Read/Write<br>The user ID of the person resolving the issue.                          |
| <b>ResolverNotes</b> | <i>String</i>                                       | Read/Write<br>Notes entered by the resolver about resolution of the Issue.            |
| <b>Severity</b>      | <i>String</i>                                       | Read/Write<br>The Issue severity - <b>Low</b> , <b>Medium</b> or <b>High</b> .        |
| <b>Status</b>        | <i>String</i>                                       | Read/Write<br>The current status of the issue.                                        |

| Attribute      | Type           | Notes                                                                                                                                                                                               |
|----------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Type</b>    | <i>Variant</i> | Read/Write<br>The Issue type - <b>Defect</b> , <b>Change</b> , <b>Issue</b> or <b>ToDo</b> .                                                                                                        |
| <b>Version</b> | <i>String</i>  | Read/Write<br>The version associated with the issue. Note that this method is only available through a Dispatch interface.<br><br><code>Object ob = Issue;</code><br><code>Print ob.Version;</code> |

**Issue Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current Issue object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.         |


**19.2.2.5.6 Metric Class**

A *Metric* is a named item with a weighting that can be associated with an element for purposes of building metrics about the model. Metrics are accessed through the Element *Metrics* collection.

**Associated table in .EAP file:** *t\_objectmetrics*

**Metric Attributes:**

| Attribute         | Type                              | Notes                                  |
|-------------------|-----------------------------------|----------------------------------------|
| <b>Name</b>       | <i>String</i>                     | Read/Write<br>The name of the metric.  |
| <b>Notes</b>      | <i>String</i>                     | Read/Write<br>Notes about this metric. |
| <b>ObjectType</b> | <a href="#"><i>ObjectType</i></a> | Read only                              |

| Attribute     | Type                                                                              | Notes                                                                     |
|---------------|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------|
|               |  | Distinguishes objects referenced through a Dispatch interface.            |
| <b>Type</b>   | <i>String</i>                                                                     | Read/Write<br>The metric type.                                            |
| <b>Weight</b> | <i>Long</i>                                                                       | Read/Write<br>A user-defined weighting for estimation or metric purposes. |

**Metric Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current Metric object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.        |

**19.2.2.5.7 Requirement Class**

An *Element Requirement* object holds information about the responsibilities of an element in the context of the model. Requirements can be accessed using the *Element Requirements* collection.

Associated table in .EAP file: *t\_objectrequires*

**Requirement Attributes:**

| Attribute         | Type          | Notes                                                                   |
|-------------------|---------------|-------------------------------------------------------------------------|
| <b>Difficulty</b> | <i>String</i> | Read/Write<br>The estimated difficulty of implementing the requirement. |
| <b>LastUpdate</b> | <i>Date</i>   | Read/Write<br>The date the requirement was last updated.                |

| Attribute            | Type                                                | Notes                                                                                                                                                                                                                                                  |
|----------------------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>          | <i>String</i>                                       | Read/Write<br>The requirement itself.                                                                                                                                                                                                                  |
| <b>Notes</b>         | <i>String</i>                                       | Read/Write<br>Further notes on the requirement.                                                                                                                                                                                                        |
| <b>ObjectType</b>    | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                                            |
| <b>ParentID</b>      | <i>Long</i>                                         | Read only<br>The <i>ElementID</i> of the element to which this requirement applies.                                                                                                                                                                    |
| <b>Priority</b>      | <i>String</i>                                       | Read/Write<br>The assigned priority of the requirement.                                                                                                                                                                                                |
| <b>RequirementID</b> | <i>Long</i>                                         | Read only<br>A local ID for this requirement.                                                                                                                                                                                                          |
| <b>Stability</b>     | <i>String</i>                                       | Read/Write<br>The estimated stability of the requirement.<br><br>This is an indication of the probability of the requirement - or understanding of the requirement - changing. High stability indicates a low probability of the requirement changing. |
| <b>Status</b>        | <i>String</i>                                       | Read/Write<br>The current status of the requirement.                                                                                                                                                                                                   |
| <b>Type</b>          | <i>String</i>                                       | Read/Write<br>The requirement type.                                                                                                                                                                                                                    |

**Requirement Methods:**

| Method                 | Type          | Notes                                                                                                                                                                         |
|------------------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i> | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error |

| Method           | Type           | Notes                                                                                                                                                                               |
|------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  |                | occurs.                                                                                                                                                                             |
| <b>Update ()</b> | <i>Boolean</i> | Updates the current Requirement object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

#### 19.2.2.5.8 Resource Class

An Element *Resource* is a named person/task pair with timing constraints and percent complete indicators. Use this to manage the work associated with delivering an Element.

Associated table in .EAP file: *t\_objectresources*

#### Resource Attributes

| Attribute            | Type          | Notes                                                                                    |
|----------------------|---------------|------------------------------------------------------------------------------------------|
| <b>ActualHours</b>   | <i>Long</i>   | Read/Write<br>The time already expended on the task, in hours, days or other units.      |
| <b>DateEnd</b>       | <i>Date</i>   | Read/Write<br>The expected end date.                                                     |
| <b>DateStart</b>     | <i>Date</i>   | Read/Write<br>The date to start work.                                                    |
| <b>ExpectedHours</b> | <i>Long</i>   | Read/Write<br>The total expected time the task might run, in hours, days or other units. |
| <b>History</b>       | <i>String</i> | Read/Write<br>Gets or sets history text.                                                 |
| <b>Name</b>          | <i>String</i> | Read/Write<br>The name of the resource (for example, a person's name).                   |
| <b>Notes</b>         | <i>String</i> | Read/Write<br>Descriptive notes.                                                         |

| Attribute              | Type                                 | Notes                                                                                          |
|------------------------|--------------------------------------|------------------------------------------------------------------------------------------------|
| <b>ObjectType</b>      | <a href="#">ObjectType</a><br>[2419] | Read only<br>Distinguishes objects referenced through a Dispatch interface.                    |
| <b>PercentComplete</b> | <i>Long</i>                          | Read/Write<br>The current percent complete figure.                                             |
| <b>Role</b>            | <i>String</i>                        | Read/Write<br>The role the resource plays in implementing the element.                         |
| <b>Time</b>            | <i>Long</i>                          | Read/Write<br>The time expected to complete the task; a numeric indicating the number of days. |

**Resource Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current Resource object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.       |

**19.2.2.5.9 Risk Class**

A *Risk* object represents a named risk associated with an element, it is used for project management purposes. Risks can be accessed through the Element *Risks* collection.

**Associated table in .EAP file:** *t\_objectrisks*

**Risk Attributes:**

| Attribute         | Type                                                | Notes                                                                       |
|-------------------|-----------------------------------------------------|-----------------------------------------------------------------------------|
| <b>Name</b>       | <i>String</i>                                       | Read/Write<br>The name of the risk.                                         |
| <b>Notes</b>      | <i>String</i>                                       | Read/Write<br>Further notes describing the risk.                            |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface. |
| <b>Type</b>       | <i>String</i>                                       | Read/Write<br>The risk type associated with this element.                   |
| <b>Weight</b>     | <i>Long</i>                                         | Read/Write<br>A weighting for estimation or metric purposes.                |

**Risk Methods:**

| Method                 | Type           | Notes                                                                                                                                                                             |
|------------------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current Risk object after modification or appending a new item.<br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.           |

**19.2.2.5.10 Scenario Class**

A *Scenario* corresponds to a Collaboration or Use Case instance. Each Scenario is a path of execution through the logic of a Use Case. Scenarios can be added to using the Element *Scenarios* collection.

**Associated table in .EAP file:** *t\_objectscenarios*

**Scenario Attributes:**

| Attribute   | Type          | Notes      | See also |
|-------------|---------------|------------|----------|
| <b>Name</b> | <i>String</i> | Read/Write |          |



| Attribute           | Type                                     | Notes                                                                                                                                                                                                        | See also                                |
|---------------------|------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
|                     |                                          | The Scenario name.                                                                                                                                                                                           |                                         |
| <b>Notes</b>        | <i>String</i>                            | Read/Write<br>A description of the Scenario, usually containing the steps to execute the scenario.                                                                                                           |                                         |
| <b>ObjectType</b>   | <a href="#">ObjectType</a> [2419]        | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                                                                                  |                                         |
| <b>ScenarioGUID</b> | <i>String</i>                            | Read/Write<br>A unique ID for the Scenario, used to identify the Scenario unambiguously within a model.                                                                                                      |                                         |
| <b>Steps</b>        | <i>Collection of ScenarioStep</i> [2502] | Read only<br>A collection of step objects for this Scenario.<br>Use the <i>AddNew</i> and <i>Delete</i> functions to manage steps. <i>AddNew</i> passes the step name and "1" as the type for an actor step. | <a href="#">Collection Class</a> [2427] |
| <b>Type</b>         | <i>String</i>                            | Read/Write<br>The scenario type (for example, <i>Basic Path</i> ).                                                                                                                                           |                                         |
| <b>Weight</b>       | <i>Long</i>                              | Read/Write<br>Currently used to position scenarios in the scenario list (that is, <i>List Position</i> ).                                                                                                    |                                         |
| <b>XMLContent</b>   | <i>String</i>                            | Read/Write<br>A structured field that can contain scenario details in XML format. It is recommended that you use the <i>Steps</i> collection to read or modify this field.                                   |                                         |

**Scenario Methods:**

| Method                 | Type          | Notes                                                                                                                                                                                 |
|------------------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i> | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |

| Method           | Type           | Notes                                                                                                                                                                           |
|------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Update ()</b> | <i>Boolean</i> | Update the current Scenario object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

#### 19.2.2.5.11 ScenarioExtension Class

##### ScenarioExtension Attributes:

| Attribute            | Type                                                  | Notes                                                                                                                                                        |
|----------------------|-------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ExtensionGUID</b> | <i>String</i>                                         | Read/Write<br>A unique GUID for this Extension.                                                                                                              |
| <b>Join</b>          | <i>String</i>                                         | Read/Write<br>The GUID of the step where this Extension rejoins the Scenario.                                                                                |
| <b>JoiningStep</b>   | <a href="#">ScenarioStep</a><br><small>[2502]</small> | Read only<br>The actual step where this Extension rejoins the Scenario, if any.                                                                              |
| <b>Level</b>         | <i>String</i>                                         | Read only<br>The number of this Extension as shown in the scenario editor. This is derived from the value of <i>Pos</i> for this object and the owning step. |
| <b>Name</b>          | <i>String</i>                                         | Read/Write<br>The Extension name. This should match the name of the linked scenario.                                                                         |
| <b>ObjectType</b>    | <a href="#">ObjectType</a><br><small>[2419]</small>   | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                                  |
| <b>Pos</b>           | <i>Long</i>                                           | Read/Write<br>The position of the Extension in the Extensions list.                                                                                          |
| <b>Scenario</b>      | <a href="#">Scenario</a><br><small>[2499]</small>     | Read only<br>The scenario that is executed as an alternative path for this Extension.                                                                        |

**ScenarioExtension Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                            |
|------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.            |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current <i>ScenarioExtension</i> object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

**19.2.2.5.12 ScenarioStep Class****ScenarioStep Attributes:**

| Attribute            | Type                                   | Notes                                                                                                                                                                                                                                  | See also                                                                                            |
|----------------------|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| <b>Extensions</b>    | <i>Collection of ScenarioExtension</i> | Read only<br><br>A collection of <i>ScenarioExtension</i> objects that specify how the scenario is extended from this step. The arguments to <i>AddNew</i> should match the name and GUID of the alternative scenario being linked to. | <a href="#">Collection</a> <sup>[2427]</sup><br><a href="#">ScenarioExtension</a> <sup>[2501]</sup> |
| <b>Level</b>         | <i>String</i>                          | Read only<br><br>The number of this Step as shown in the scenario editor. This is derived from the value of <i>Pos</i> .                                                                                                               |                                                                                                     |
| <b>Link</b>          | <i>String</i>                          | Read/Write<br><br>The GUID of a Use Case that is relevant to this step.                                                                                                                                                                |                                                                                                     |
| <b>LinkedElement</b> | <i>Element</i>                         | Read only<br><br>The actual element specified by Link, if any.                                                                                                                                                                         | <a href="#">Element</a> <sup>[2478]</sup>                                                           |
| <b>Name</b>          | <i>String</i>                          | Read/Write<br><br>The sSep name.                                                                                                                                                                                                       |                                                                                                     |
| <b>ObjectType</b>    | <i>ObjectType</i>                      | Read only<br><br>Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                        | <a href="#">ObjectType</a> <sup>[2419]</sup>                                                        |

| Attribute              | Type                         | Notes                                                                                                    | See also                                                                                                                   |
|------------------------|------------------------------|----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
|                        |                              |                                                                                                          |                                                                                                                            |
| <b>Pos</b>             | <i>Long</i>                  | Read/Write<br>The position of the Step in the Scenario Step list.                                        |                                                                                                                            |
| <b>Results</b>         | <i>String</i>                | Read/Write<br>Any results that are given from this step.                                                 |                                                                                                                            |
| <b>State</b>           | <i>String</i>                | Read/Write<br>A description of the state the system enters when this Step is executed.                   |                                                                                                                            |
| <b>StepGUID</b>        | <i>String</i>                | Read/Write<br>A unique GUID for this Step.                                                               |                                                                                                                            |
| <b>StepType</b>        | <i>ScenarioStepType</i>      | Read/Write<br>Identifies whether this step is being performed by a user or the system.                   | <a href="#">ScenarioStepType</a> [2422]                                                                                    |
| <b>Uses</b>            | <i>String</i>                | Read/Write<br>The input and requirements that are relevant to this step.                                 |                                                                                                                            |
| <b>UsesElementList</b> | <i>Collection of Element</i> | Read only<br>Indicates that the Structured Specification tab <b>Uses</b> field is a linked element list. | <a href="#">Collection</a> [2427]<br><a href="#">Element</a> [2478]<br><a href="#">Set Up Scenario Specification</a> [855] |

**ScenarioStep Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current <i>ScenarioStep</i> object after modification or appending a new item.                                                                                            |

| Method | Type | Notes                                                                                     |
|--------|------|-------------------------------------------------------------------------------------------|
|        |      | If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

### 19.2.2.5.13 TaggedValue Class

A *TaggedValue* is a named property and value associated with an element. Tagged Values can be accessed through the *TaggedValues* collection.

Associated table in .EAP file: *t\_objectproperties*

#### TaggedValue Attributes:

| Attribute           | Type                                 | Notes                                                                                                                                                                                   |
|---------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ElementID</b>    | <i>Long</i>                          | Read/Write<br>The local ID of the associated element.                                                                                                                                   |
| <b>FQName</b>       | <i>String</i>                        | Read only<br>The fully-qualified name of the tag.                                                                                                                                       |
| <b>Name</b>         | <i>String</i>                        | Read/Write<br>The name of the tag.                                                                                                                                                      |
| <b>Notes</b>        | <i>String</i>                        | Read/Write<br>Further descriptive notes about this tag.<br>If <b>Value</b> (below) is set to " <b>&lt;memo&gt;</b> ", then <b>Notes</b> should contain the actual Tagged Value content. |
| <b>ObjectType</b>   | <a href="#">ObjectType</a><br>[2419] | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                                                             |
| <b>PropertyGUID</b> | <i>String</i>                        | Read/Write<br>The global ID of the tag.                                                                                                                                                 |
| <b>PropertyID</b>   | <i>Long</i>                          | Read only<br>The local ID of the tag.                                                                                                                                                   |
| <b>Value</b>        | <i>String</i>                        | Read/Write<br>The value assigned to this tag.                                                                                                                                           |

| Attribute | Type | Notes                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           |      | <p>This field has a 255 character limit. If the value is greater than 255 characters long, set the value to "&lt;memo&gt;" and insert the body of text in the <b>Notes</b> attribute (above).</p> <p>When reading existing Tagged Values, if <b>Value</b> = "&lt;memo&gt;" then the developer should read the actual body of text from the <b>Notes</b> attribute.</p> |

#### TaggedValue Methods:

| Method                                                     | Type           | Notes                                                                                                                                                                                                                                                                            |
|------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetAttribute</b><br>(string propName)                   | <i>String</i>  | <p>Returns the text of a single named property within a structured Tagged Value.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>propName: String - the name of the property for which the text is being returned</li> </ul>                                       |
| <b>GetLastError()</b>                                      | <i>String</i>  | <p>Returns a string value describing the most recent error that occurred in relation to this object.</p> <p>This function is rarely used as an exception is thrown when an error occurs.</p>                                                                                     |
| <b>HasAttributes()</b>                                     | <i>Boolean</i> | <p>Returns true if the Tagged Value is a structured Tagged Value with one or more properties.</p>                                                                                                                                                                                |
| <b>SetAttribute</b><br>(string propName, string propValue) | <i>Boolean</i> | <p>Sets the text of a single named property within a structured Tagged Value.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>propName: String - the name of the property for which the text is being set</li> <li>propValue: the value of the property</li> </ul> |
| <b>Update()</b>                                            | <i>Boolean</i> | <p>Updates the current TaggedValue object after modification or appending a new item.</p> <p>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.</p>                                                                                       |

#### 19.2.2.5.14 Test Class

A *Test* is a single Test Case applied to an element. Tests are added and accessed through the Element *Tests* collection.

**Associated table in .EAP file:** *t\_objecttests*.

#### Test Attributes:

| Attribute                 | Type                                                | Notes                                                                                                                                                                                                            |
|---------------------------|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AcceptanceCriteria</b> | <i>String</i>                                       | Read/Write<br>The acceptance criteria for successful execution.                                                                                                                                                  |
| <b>CheckedBy</b>          | <i>String</i>                                       | Read/Write<br>User ID of the person confirming the results.                                                                                                                                                      |
| <b>Class</b>              | <i>Long</i>                                         | Read/Write<br>The test Class:<br><ul style="list-style-type: none"> <li>1 = Unit Test</li> <li>2 = Integration Test</li> <li>3 = System Test</li> <li>4 = Acceptance Test</li> <li>5 = Scenario Test.</li> </ul> |
| <b>DateRun</b>            | <i>Date</i>                                         | Read/Write<br>The date the test was last run.                                                                                                                                                                    |
| <b>Input</b>              | <i>String</i>                                       | Read/Write<br>Input data for the test.                                                                                                                                                                           |
| <b>Name</b>               | <i>String</i>                                       | Read/Write<br>The test name.                                                                                                                                                                                     |
| <b>Notes</b>              | <i>String</i>                                       | Read/Write<br>Detailed notes about test to be carried out.                                                                                                                                                       |
| <b>ObjectType</b>         | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                                                                                      |

| Attribute          | Type           | Notes                                                         |
|--------------------|----------------|---------------------------------------------------------------|
| <b>RunBy</b>       | <i>String</i>  | Read/Write<br>.The user ID of the person conducting the test. |
| <b>Status</b>      | <i>String</i>  | Read/Write<br>. Current status of test.                       |
| <b>TestResults</b> | <i>Variant</i> | Read/Write<br>. Results of test.                              |
| <b>Type</b>        | <i>String</i>  | Read/Write<br>The test type, such as Load or Regression.      |

**Test Methods:**

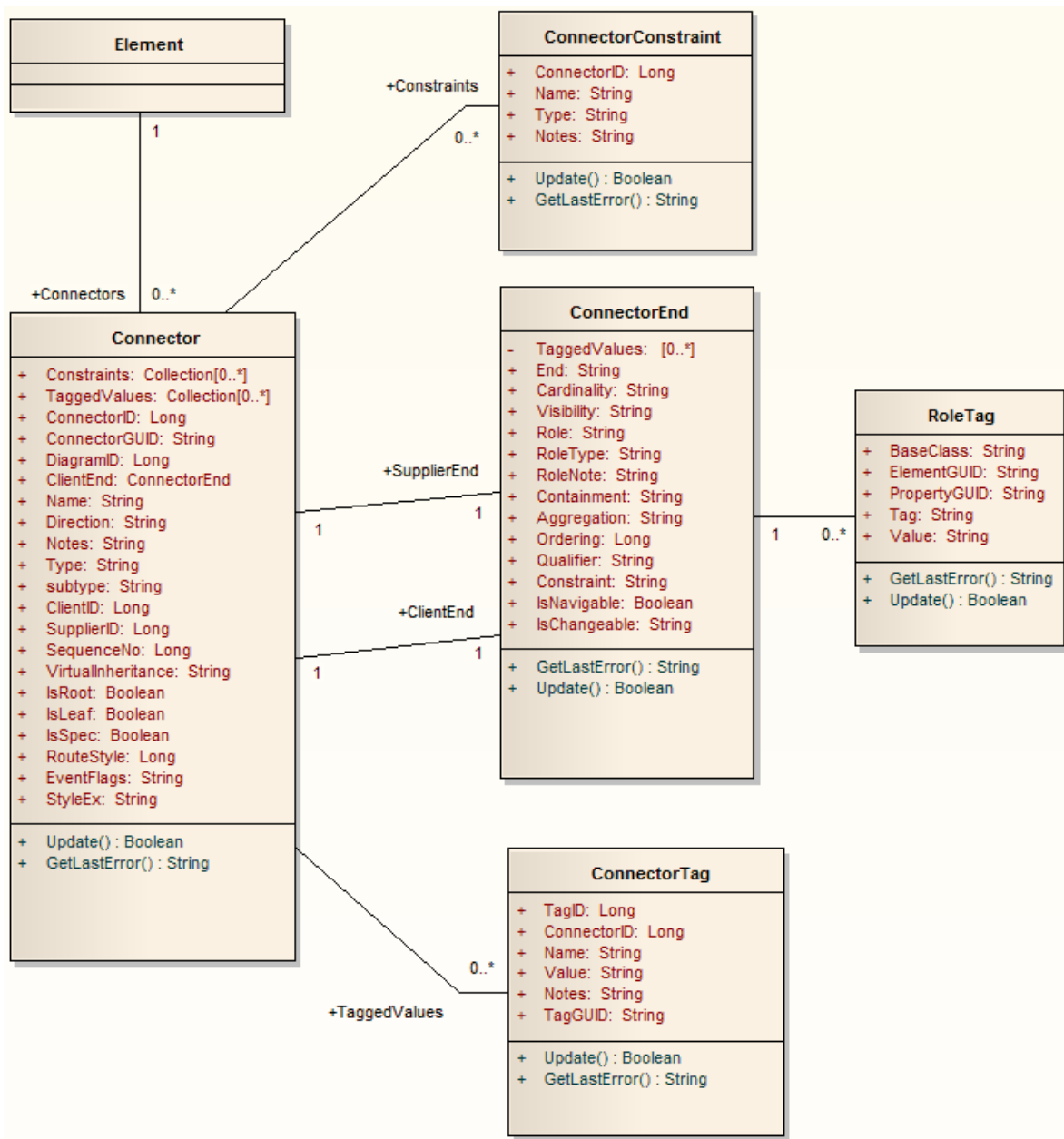
| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current Test object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.           |

**19.2.2.6 Connector Package**

The *Connector* package details how connectors between elements are accessed and managed.

**Example**





### 19.2.2.6.1 Connector Class

Associated table in .EAP file: `t_connector`

#### Reference

| Topic                | Detail                                                                                                                                                                                                                                                                                                          | See also |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>General Usage</b> | <p>A <i>Connector</i> object represents the various kinds of connectors between UML elements; it is accessed from either the <i>Client</i> or <i>Supplier</i> element, using the <i>Connectors</i> collection of that element</p> <p>When creating a new connector you must assign it a valid type from the</p> |          |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>following list:</p> <ul style="list-style-type: none"> <li>• Aggregation</li> <li>• Assembly</li> <li>• Association</li> <li>• Collaboration</li> <li>• CommunicationPath</li> <li>• Connector</li> <li>• ControlFlow</li> <li>• Delegate</li> <li>• Dependency</li> <li>• Deployment</li> <li>• ERLink</li> <li>• Generalization</li> <li>• InformationFlow</li> <li>• Instantiation</li> <li>• InterruptFlow</li> <li>• Manifest</li> <li>• Nesting</li> <li>• NoteLink</li> <li>• ObjectFlow</li> <li>• Package</li> <li>• Realization</li> <li>• Sequence</li> <li>• StateFlow</li> <li>• TemplateBinding</li> <li>• UseCase</li> </ul> |          |

**Connector Attributes:**

| Attribute        | Type                                           | Notes                                                                     | See also |
|------------------|------------------------------------------------|---------------------------------------------------------------------------|----------|
| <b>Alias</b>     | <i>String</i>                                  | Read/Write<br>An optional alias for this connector.                       |          |
| <b>ClientEnd</b> | <a href="#">ConnectorEnd</a> <sup>[2515]</sup> | Read only<br>A pointer to the <i>ConnectorEnd</i> object representing the |          |

| Attribute               | Type                              | Notes                                                                                                                                                                                                                                                                          | See also                                         |
|-------------------------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
|                         |                                   | source end of the relationship.                                                                                                                                                                                                                                                |                                                  |
| <b>ClientID</b>         | <i>Long</i>                       | Read/Write<br>The <i>ElementID</i> of the element at the source end of this connector.                                                                                                                                                                                         |                                                  |
| <b>Color</b>            | <i>Long</i>                       | Read/Write<br>Sets the color of the connector.                                                                                                                                                                                                                                 |                                                  |
| <b>ConnectorGUID</b>    | <i>Variant</i>                    | Read only<br>A system generated, globally unique ID for the current connector.                                                                                                                                                                                                 |                                                  |
| <b>ConnectorID</b>      | <i>Long</i>                       | Read only<br>A system generated local identifier for the current connector.                                                                                                                                                                                                    |                                                  |
| <b>Constraints</b>      | <a href="#">Collection</a> [2427] | Read only<br>A collection of <b>constraint</b> objects.                                                                                                                                                                                                                        | <a href="#">Constraint</a> [2476]                |
| <b>ConveyedItems</b>    | <a href="#">Collection</a> [2427] | Read only<br>Returns a collection of elements that have been conveyed.                                                                                                                                                                                                         |                                                  |
| <b>CustomProperties</b> | <a href="#">Collection</a> [2427] | Read only<br>Returns a collection of advanced properties associated with an element in the form of <b>CustomProperty</b> objects.                                                                                                                                              | <a href="#">CustomProperty Collection</a> [2399] |
| <b>DiagramID</b>        | <i>Long</i>                       | Read/Write<br>The <i>DiagramID</i> of the connector.                                                                                                                                                                                                                           |                                                  |
| <b>Direction</b>        | <i>String</i>                     | Read/Write<br>The connector direction, which can be set to one of the following: <ul style="list-style-type: none"> <li>• <b>Unspecified</b></li> <li>• <b>Bi-Directional</b></li> <li>• <b>Source -&gt; Destination</b></li> <li>• <b>Destination -&gt; Source</b></li> </ul> | <a href="#">Supported Attributes</a> [1308]      |

| Attribute         | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                         | See also |
|-------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                   |                | If the connector is non-navigable, set the <b>sourceNavigability</b> and/or <b>targetNavigability</b> attributes.                                                                                                                                                                                                                                                                                                             |          |
| <b>EndPointX</b>  | <i>Long</i>    | Read/Write<br>The x-coordinate of the connector's end point.<br>Connector end points are specified in Cartesian coordinates with the origin to the top left of the screen.                                                                                                                                                                                                                                                    |          |
| <b>EndPointY</b>  | <i>Long</i>    | Read/Write<br>The y-coordinate of the connector's end point.<br>Connector end points are specified in Cartesian coordinates with the origin to the top left of the screen.                                                                                                                                                                                                                                                    |          |
| <b>EventFlags</b> | <i>String</i>  | Read/Write<br>A structure to hold a variety of flags concerned with event signaling on messages.                                                                                                                                                                                                                                                                                                                              |          |
| <b>IsLeaf</b>     | <i>Boolean</i> | Read/Write<br>A flag indicating that the connector is a <i>leaf</i> .                                                                                                                                                                                                                                                                                                                                                         |          |
| <b>IsRoot</b>     | <i>Boolean</i> | Read/Write<br>A flag indicating that the connector is a <i>root</i> .                                                                                                                                                                                                                                                                                                                                                         |          |
| <b>IsSpec</b>     | <i>Boolean</i> | Read/Write.<br>A flag indicating that the connector is a specification.                                                                                                                                                                                                                                                                                                                                                       |          |
| <b>MetaType</b>   | <i>String</i>  | Read only<br>The connector's domain-specific meta type, as defined by an applied stereotype from an MDG Technology                                                                                                                                                                                                                                                                                                            |          |
| <b>MiscData</b>   | <i>String</i>  | Read only<br>This low-level property returns an array providing information about the contents of the <b>PData</b> fields; these database fields are not documented and developers must gain understanding of these fields through their own endeavors to use this property.<br><b>MiscData</b> is zero based, therefore: <ul style="list-style-type: none"> <li>• <b>MiscData(0)</b> corresponds to <b>PData1</b></li> </ul> |          |

| Attribute          | Type                                                           | Notes                                                                                                                                                                                                                                                                                              | See also |
|--------------------|----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                    |                                                                | <ul style="list-style-type: none"> <li>• <b>MiscData(1) to PData2</b></li> </ul> <p>and so on.</p>                                                                                                                                                                                                 |          |
| <b>Name</b>        | <i>String</i>                                                  | Read/Write<br>The connector name.                                                                                                                                                                                                                                                                  |          |
| <b>Notes</b>       | <i>String</i>                                                  | Read/Write<br>Descriptive notes about the connector.                                                                                                                                                                                                                                               |          |
| <b>ObjectType</b>  | <a href="#">ObjectTyp</a><br><small>e<sup>[2410]</sup></small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                                                                                        |          |
| <b>Properties</b>  | <a href="#">Properties</a><br><small>[2410]</small>            | Returns a list of specialized properties applicable to the connector that might not be available using the automation model.<br><br>The properties are purposely undocumented because of their obscure nature and because they are subject to change as progressive enhancements are made to them. |          |
| <b>RouteStyle</b>  | <i>Long</i>                                                    | Read/Write<br>The route style.                                                                                                                                                                                                                                                                     |          |
| <b>SequenceNo</b>  | <i>Long</i>                                                    | Read/Write<br>The <i>SequenceNo</i> of the connector.                                                                                                                                                                                                                                              |          |
| <b>StartPointX</b> | <i>Long</i>                                                    | Read/Write<br>The x-coordinate of the connector's start point.<br><br>Connector end points are specified in Cartesian coordinates with the origin to the top left of the screen.                                                                                                                   |          |
| <b>StartPointY</b> | <i>Long</i>                                                    | Read/Write<br>The y-coordinate of the connector's start point.<br><br>Connector end points are specified in Cartesian coordinates with the origin to the top left of the screen.                                                                                                                   |          |
| <b>StateFlags</b>  | <i>String</i>                                                  | Read/Write.                                                                                                                                                                                                                                                                                        |          |

| Attribute               | Type                                           | Notes                                                                                                                    | See also                                     |
|-------------------------|------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
|                         |                                                | A structure to hold a variety of flags concerned with State signaling on messages; the list is delimited by semi-colons. |                                              |
| <b>Stereotype</b>       | <i>String</i>                                  | Read/Write<br>Sets or gets the stereotype for this connector end.                                                        |                                              |
| <b>StereotypeEx</b>     | <i>String</i>                                  | Read/Write<br>All the applied stereotypes of the connector in a comma-separated list.                                    |                                              |
| <b>StyleEx</b>          | <i>String</i>                                  | Read/Write<br>Advanced style settings; reserved for the use of Sparx Systems.                                            |                                              |
| <b>Subtype</b>          | <i>String</i>                                  | Read/Write<br>A possible subtype to refine the meaning of the connector.                                                 |                                              |
| <b>SupplierEnd</b>      | <a href="#">ConnectorEnd</a> <sup>[2515]</sup> | Read only<br>A pointer to the <i>ConnectorEnd</i> object representing the target end of the relationship.                |                                              |
| <b>SupplierID</b>       | <i>Long</i>                                    | Read/Write<br>The <i>ElementID</i> of the element at the target end of this connector.                                   |                                              |
| <b>TaggedValues</b>     | <i>Collection</i>                              | Read only<br>The collection of <i>ConnectorTag</i> objects.                                                              |                                              |
| <b>TransitionAction</b> | <i>String</i>                                  | Read/Write<br>See the <i>Transition</i> topic for appropriate values.                                                    | <a href="#">Transition</a> <sup>[1260]</sup> |
| <b>TransitionEvent</b>  | <i>String</i>                                  | Read/Write<br>See the <i>Transition</i> topic for appropriate values.                                                    |                                              |
| <b>TransitionGuard</b>  | <i>String</i>                                  | Read/Write<br>See the <i>Transition</i> topic for appropriate values.                                                    |                                              |

| Attribute                 | Type          | Notes                                                                                                         | See also |
|---------------------------|---------------|---------------------------------------------------------------------------------------------------------------|----------|
| <b>Type</b>               | <i>String</i> | Read/Write<br>The connector type; valid types are held in the <i>t_connectortypes</i> table in the .EAP file. |          |
| <b>VirtualInheritance</b> | <i>String</i> | Read/Write<br>For <i>Generalization</i> , indicates if the inheritance is virtual.                            |          |
| <b>Width</b>              | <i>Long</i>   | Read/Write<br>Specifies the width of the connector.                                                           |          |

**Connector Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                  |
|------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used, as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current <i>ConnectorObject</i> after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

**19.2.2.6.2 ConnectorConstraint Class**

A *ConnectorConstraint* holds information about special conditions that apply to a connector. It is accessed through the *Connector Constraints* collection.

**Associated table in .EAP file:** *t\_connectorconstraints*

**ConnectorConstraint Attributes:**

| Attribute          | Type          | Notes                                                     |
|--------------------|---------------|-----------------------------------------------------------|
| <b>ConnectorID</b> | <i>Long</i>   | Read/Write<br>A local ID value (long) - system generated. |
| <b>Name</b>        | <i>String</i> | Read/Write<br>The constraint name.                        |

| Attribute         | Type                                                | Notes                                                                       |
|-------------------|-----------------------------------------------------|-----------------------------------------------------------------------------|
|                   |                                                     |                                                                             |
| <b>Notes</b>      | <i>String</i>                                       | Read/Write<br>Notes about this constraint.                                  |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface. |
| <b>Type</b>       | <i>String</i>                                       | Read/Write<br>The constraint type.                                          |

**ConnectorConstraint Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                             |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.             |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current <i>ConnectorConstraint</i> object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

**19.2.2.6.3 ConnectorEnd Class**

A *ConnectorEnd* contains information about a single end of a connector. A *ConnectorEnd* is accessed from the connector as either the *ClientEnd* or *SupplierEnd*.

**Associated table in .EAP file:** *derived from t\_connector*

**ConnectorEnd Attributes:**

| Attribute          | Type        | Notes                                                                                                                                                      | See also |
|--------------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Aggregation</b> | <i>Long</i> | Read/Write<br><br>The type of Aggregation as it applies to this end; valid values are:<br><br><b>0</b> = None<br><b>1</b> = Shared<br><b>2</b> = Composite |          |



| Attribute                  | Type           | Notes                                                                                                                 | See also |
|----------------------------|----------------|-----------------------------------------------------------------------------------------------------------------------|----------|
|                            |                |                                                                                                                       |          |
| <b>Alias</b>               | <i>String</i>  | Read/Write<br>An optional alias for this connector end.                                                               |          |
| <b>AllowDuplicat<br/>s</b> | <i>Boolean</i> | Read/Write<br>For multiplicities greater than <b>1</b> , indicates that duplicate entries are possible.               |          |
| <b>Cardinality</b>         | <i>String</i>  | Read/Write<br>The cardinality associated with this end.                                                               |          |
| <b>Constraint</b>          | <i>String</i>  | Read/Write<br>A constraint that can be applied to this connector end.                                                 |          |
| <b>Containment</b>         | <i>String</i>  | Read/Write<br>The containment type applied to this connector end.                                                     |          |
| <b>Derived</b>             | <i>Boolean</i> | Read/Write<br>Indicates that the value of this end is derived.                                                        |          |
| <b>DerivedUnion</b>        | <i>Boolean</i> | Read/Write<br>Indicates the value of this role derived from the union of all roles that subset this.                  |          |
| <b>End</b>                 | <i>String</i>  | Read only<br>The end this <i>ConnectorEnd</i> object applies to - <b>Client</b> or <b>Supplier</b> .                  |          |
| <b>IsChangeable</b>        | <i>String</i>  | Read/Write<br>Flag indicating whether this end is changeable or not - <b>frozen</b> , <b>addOnly</b> or <b>none</b> . |          |
| <b>IsNavigable</b>         | <i>Boolean</i> | Read/Write<br>A flag indicating this end is navigable from the other end.                                             |          |
| <b>Navigable</b>           | <i>String</i>  | Read/Write<br>Indicates whether this role of an association is navigable                                              |          |

| Attribute                | Type                                                | Notes                                                                                                                 | See also                                               |
|--------------------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
|                          |                                                     | from the opposite classifier - <b>Navigable, Non-Navigable</b> or <b>Unspecified</b> .                                |                                                        |
| <b>ObjectType</b>        | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                           |                                                        |
| <b>Ordering</b>          | <i>Long</i>                                         | Read/Write<br>Ordering for this connector end.                                                                        |                                                        |
| <b>OwnedByClassifier</b> | <i>Boolean</i>                                      | Read/Write<br>Indicates that this Association end corresponds to an attribute on the opposite end of the Association. |                                                        |
| <b>Qualifier</b>         | <i>String</i>                                       | Read/Write<br>A qualifier that can apply to the connector end.                                                        |                                                        |
| <b>Role</b>              | <i>String</i>                                       | Read/Write<br>The connector end role.                                                                                 |                                                        |
| <b>RoleNote</b>          | <i>String</i>                                       | Read/Write<br>Notes associated with the role of this connector end.                                                   |                                                        |
| <b>RoleType</b>          | <i>String</i>                                       | Read/Write<br>The role type applied to this end of the connector.                                                     |                                                        |
| <b>Stereotype</b>        | <i>String</i>                                       | Read/Write<br>Sets or gets the stereotype for this connector end.                                                     |                                                        |
| <b>StereotypeEx</b>      | <i>String</i>                                       | Read/Write<br>All the applied stereotypes of the connector end in a comma-separated list.                             |                                                        |
| <b>TaggedValues</b>      | <i>Private</i>                                      | Read only<br>A collection of <i>RoleTag</i> objects.                                                                  | <a href="#">RoleTag Class</a><br><small>[2520]</small> |
| <b>Visibility</b>        | <i>String</i>                                       | Read/Write                                                                                                            |                                                        |

| Attribute | Type | Notes                                                                                                                | See also |
|-----------|------|----------------------------------------------------------------------------------------------------------------------|----------|
|           |      | The Scope associated with this connector end - <b>Public</b> , <b>Private</b> , <b>Protected</b> or <b>Package</b> . |          |

#### ConnectorEnd Methods:

| Method                 | Type           | Notes                                                                                                                                                                                      |
|------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.      |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current <i>ConnectorEnd</i> object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

#### 19.2.2.6.4 ConnectorTag Class

A *ConnectorTag* is a Tagged Value for a connector and is accessed through the Connector *TaggedValues* collection.

Associated table in .EAP file: *t\_connectortag*

#### ConnectorTag Attributes:

| Attribute          | Type          | Notes                                                                                                                                                   |
|--------------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ConnectorID</b> | <i>Long</i>   | Read/Write<br>The local ID of the associated connector.                                                                                                 |
| <b>FQName</b>      | <i>String</i> | Read only<br>The fully qualified name of the tag.                                                                                                       |
| <b>Name</b>        | <i>String</i> | Read/Write<br>The tag or name.                                                                                                                          |
| <b>Notes</b>       | <i>String</i> | Read/Write<br>Further descriptive notes on this tag.<br>If <b>Value</b> (below) is set to " <b>&lt;memo&gt;</b> ", then <b>Notes</b> should contain the |

| Attribute         | Type                                                | Notes                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   |                                                     | actual Tagged Value content.                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                                                                                                                                                                                                                   |
| <b>TagGUID</b>    | <i>String</i>                                       | Read/Write<br>A globally unique ID for this Tagged Value.                                                                                                                                                                                                                                                                                                                                                                     |
| <b>TagID</b>      | <i>Long</i>                                         | Read only<br>A local ID to identify the Tagged Value.                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Value</b>      | <i>String</i>                                       | Read/Write<br>The value assigned to this tag.<br><br>This field has a 255 character limit. If the value is greater than 255 characters long, set the value to " <b>&lt;memo&gt;</b> " and insert the body of text in the <b>Notes</b> attribute (above).<br><br>When reading existing Tagged Values, if <b>Value = "&lt;memo&gt;"</b> then the developer should read the actual body of text from the <b>Notes</b> attribute. |

**ConnectorTag Methods:**

| Method                                                              | Type           | Notes                                                                                                                                                                                 |
|---------------------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetAttribute</b><br>(string<br>propName)                         | <i>String</i>  | Returns the text of a single named property within a structured tagged value.                                                                                                         |
| <b>GetLastError()</b>                                               | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>HasAttributes()</b>                                              | <i>Boolean</i> | Returns true if the tagged values is a structured tagged value with one or more properties.                                                                                           |
| <b>SetAttribute</b><br>(string<br>propName,<br>string<br>propValue) | <i>Boolean</i> | Sets the text of a single named property within a structured tagged value.                                                                                                            |
| <b>Update()</b>                                                     | <i>Boolean</i> | Update the current <i>ConnectorTag</i> object after modification or                                                                                                                   |

| Method | Type | Notes                                                                                                                         |
|--------|------|-------------------------------------------------------------------------------------------------------------------------------|
|        |      | <p>appending a new item.</p> <p>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.</p> |

### 19.2.2.6.5 RoleTag Class

#### Reference

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See also |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>General Usage</b> | <p>This interface provides access to the association Role Tagged Values. Each connector end has a RoleTag collection that can be accessed to add, delete and access the RoleTags.</p> <p>In code you create something that resembles the following code fragment for accessing a RoleTag in VB.NET (where <i>con</i> is a Connector Object):</p> <pre> client = con.ClientEnd client.Role = "m_client" client.Update() tag = client.TaggedValues.AddNew("tag", " value") tag.Update() tag = client.TaggedValues.AddNew("tag2", " value2") tag.Update() client.TaggedValues.Refresh() For idx = 0 To client.TaggedValues.Count - 1     tag = client.TaggedValues.GetAt(idx)     Console.WriteLine(tag.Tag)     client.TaggedValues.DeleteAt(idx, False) Next tag = Nothing </pre> |          |

#### RoleTag Attributes:

| Attribute          | Type          | Notes                                                                                                           |
|--------------------|---------------|-----------------------------------------------------------------------------------------------------------------|
| <b>BaseClass</b>   | <i>String</i> | <p>Read/Write</p> <p>Indicates the role end; set to <b>ASSOCIATION_SOURCE</b> or <b>ASSOCIATION_TARGET</b>.</p> |
| <b>ElementGUID</b> | <i>String</i> | <p>Read/Write</p> <p>The GUID of the connector with which this role tag is associated.</p>                      |
| <b>FQName</b>      | <i>String</i> | Read only                                                                                                       |

| Attribute           | Type                                                | Notes                                                                       |
|---------------------|-----------------------------------------------------|-----------------------------------------------------------------------------|
|                     |                                                     | The fully qualified name of the tag.                                        |
| <b>ObjectType</b>   | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface. |
| <b>PropertyGUID</b> | <i>String</i>                                       | Read/Write<br>A system generated GUID to identify the Tagged Value.         |
| <b>Tag</b>          | <i>String</i>                                       | Read/Write<br>The actual tag name.                                          |
| <b>Value</b>        | <i>String</i>                                       | Read/Write<br>The value associated with this tag.                           |

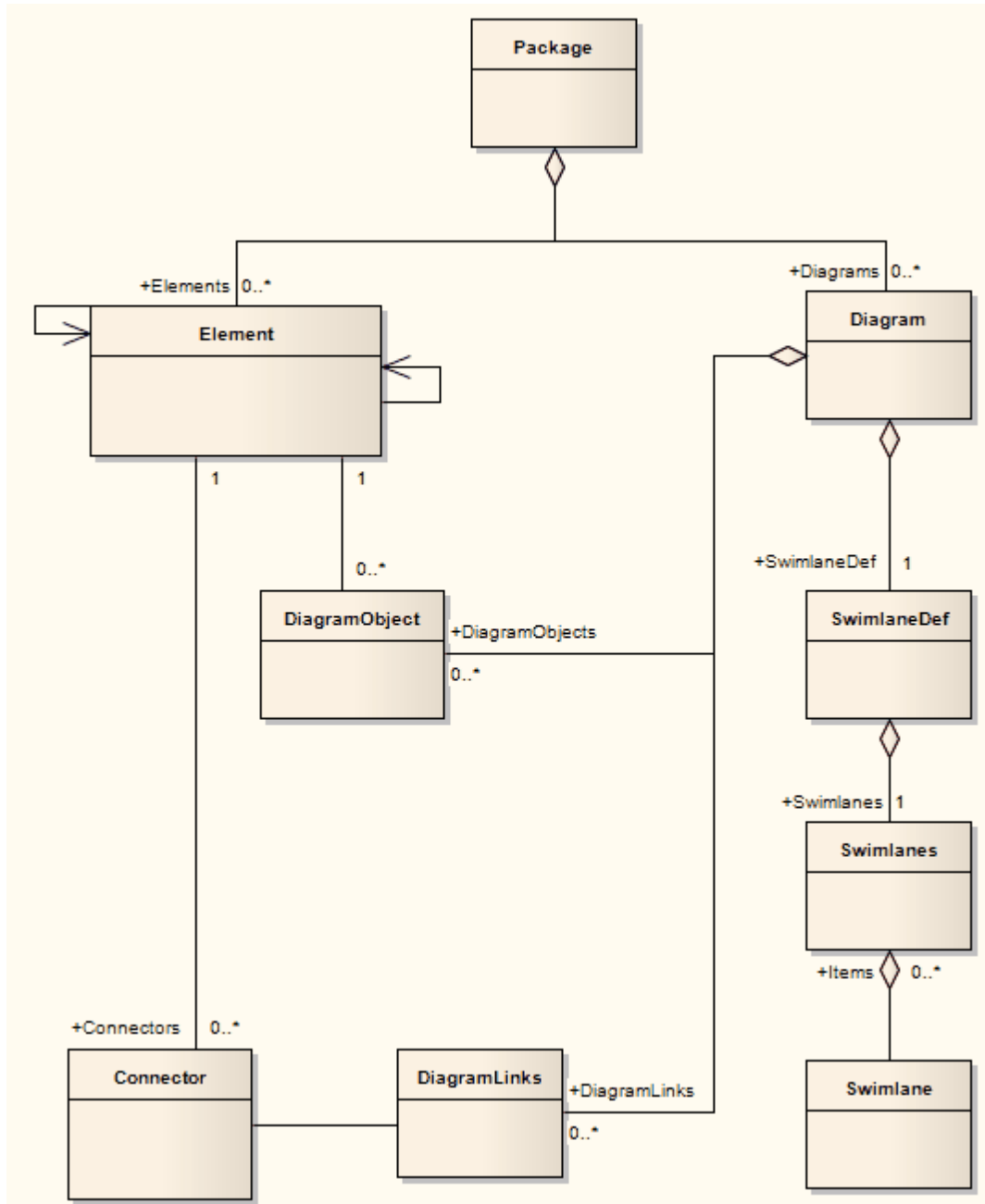
**RoleTag Methods:**

| Method                                                     | Type           | Notes                                                                                                                                                                                 |
|------------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetAttribute</b><br>(string propName)                   | <i>String</i>  | Returns the text of a single named property within a structured tagged value.                                                                                                         |
| <b>GetLastError()</b>                                      | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>HasAttributes</b><br>( )                                | <i>Boolean</i> | Returns true if the tagged values is a structured tagged value with one or more properties.                                                                                           |
| <b>SetAttribute</b><br>(string propName, string propValue) | <i>Boolean</i> | Sets the text of a single named property within a structured tagged value.                                                                                                            |
| <b>Update()</b>                                            | <i>Boolean</i> | Update the RoleTag after changes or on initial creation.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.                             |

### 19.2.2.7 Diagram Package

The *Diagram* package has information on a diagram and on *DiagramObject* and *DiagramLink*, which are the instances of elements within a diagram.

#### Example



### 19.2.2.7.1 Diagram Class

#### Topics

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See also |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>General Usage</b> | <p>A <i>Diagram</i> corresponds to a single Enterprise Architect diagram. It is accessed through the <i>Package Diagrams</i> collection and in turn contains a collection of diagram objects and diagram connectors. Adding to the <i>DiagramObjectClass</i> adds an element to the diagram (the element must already exist). When adding a new diagram, you must set the diagram type to a valid type; these are:</p> <ul style="list-style-type: none"> <li>• Activity</li> <li>• Analysis</li> <li>• Component</li> <li>• Custom</li> <li>• Deployment</li> <li>• Logical</li> <li>• Sequence</li> <li>• Statechart</li> <li>• Use Case</li> </ul> <p>Use the Analysis type for a Collaboration Diagram.</p> |          |

**Associated table in .EAP file:** *t\_diagram*

#### Diagram Attributes:

| Attribute          | Type          | Notes                                                                       | See also |
|--------------------|---------------|-----------------------------------------------------------------------------|----------|
| <b>Author</b>      | <i>String</i> | Read/Write<br>The name of the author.                                       |          |
| <b>CreatedDate</b> | <i>Date</i>   | Read/Write<br>The date the diagram was created.                             |          |
| <b>cx</b>          | <i>Long</i>   | Read/Write<br>The X dimension of the diagram (the default is <b>800</b> ).  |          |
| <b>cy</b>          | <i>Long</i>   | Read/Write<br>The Y dimension of the diagram (the default is <b>1100</b> ). |          |



| Attribute               | Type                                         | Notes                                                                                                                                                                                                                                                                                                                                                     | See also                                              |
|-------------------------|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| <b>DiagramGUID</b>      | <i>Variant</i>                               | Read/Write<br>A globally unique ID for this diagram.                                                                                                                                                                                                                                                                                                      |                                                       |
| <b>DiagramID</b>        | <i>Long</i>                                  | Read only<br>A local ID for the diagram.                                                                                                                                                                                                                                                                                                                  |                                                       |
| <b>DiagramLinks</b>     | <a href="#">Collection</a> <sup>[2427]</sup> | Read only<br>A list of <b>DiagramLink</b> objects, each containing information about the display characteristics of a connector in a diagram.<br><br>A <b>DiagramLink</b> is only created once a user modifies a connector in a diagram in some way. Until this condition has been met, default values are used and the <b>DiagramLink</b> is not in use. | <a href="#">DiagramLink Class</a> <sup>[2528]</sup>   |
| <b>DiagramObjects</b>   | <a href="#">Collection</a> <sup>[2427]</sup> | Read only<br>A collection of references to <b>DiagramObjects</b> . A <b>DiagramObject</b> is an instance of an element in a diagram, and includes size and display characteristics.                                                                                                                                                                       | <a href="#">DiagramObject Class</a> <sup>[2529]</sup> |
| <b>ExtendedStyle</b>    | <i>String</i>                                | Read/Write<br>An extended style attribute.                                                                                                                                                                                                                                                                                                                |                                                       |
| <b>HighlightImports</b> | <i>Boolean</i>                               | Read/Write<br>A flag to indicate that elements from other packages should be highlighted.                                                                                                                                                                                                                                                                 |                                                       |
| <b>IsLocked</b>         | <i>Boolean</i>                               | Read/Write<br>A flag indicating whether this diagram is locked or not.                                                                                                                                                                                                                                                                                    |                                                       |
| <b>MetaType</b>         | <i>String</i>                                | Read only<br>The diagram's domain-specific meta type, as defined by an MDG Technology.                                                                                                                                                                                                                                                                    |                                                       |
| <b>ModifiedDate</b>     | <i>Variant</i>                               | Read/Write<br>The date the diagram was last modified.                                                                                                                                                                                                                                                                                                     |                                                       |
| <b>Name</b>             | <i>String</i>                                | Read/Write<br>The diagram name.                                                                                                                                                                                                                                                                                                                           |                                                       |

| Attribute                  | Type                                              | Notes                                                                                                                                                                                                                                                  | See also |
|----------------------------|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Notes</b>               | <i>String</i>                                     | Read/Write<br>Set or retrieve notes for this diagram.                                                                                                                                                                                                  |          |
| <b>ObjectType</b>          | <a href="#">ObjectTy<br/>pe</a> <sup>[2419]</sup> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                                            |          |
| <b>Orientation</b>         | <i>String</i>                                     | Read/Write<br>The page orientation: <b>P</b> for Portrait or <b>L</b> for Landscape.                                                                                                                                                                   |          |
| <b>PackageID</b>           | <i>Long</i>                                       | Read/Write<br>The ID of the package that this diagram belongs to.                                                                                                                                                                                      |          |
| <b>ParentID</b>            | <i>Long</i>                                       | Read/Write<br>The optional ID of an element that 'owns' this diagram; for example, a Sequence diagram owned by a Use Case.                                                                                                                             |          |
| <b>Scale</b>               | <i>Long</i>                                       | Read/Write<br>The zoom scale (the default is <b>100</b> ).                                                                                                                                                                                             |          |
| <b>SelectedConnector</b>   | <a href="#">Connect<br/>or</a> <sup>[2508]</sup>  | Read/Write<br>The currently selected connector on this diagram. Null if there is no currently selected diagram.                                                                                                                                        |          |
| <b>SelectedObjects</b>     | <a href="#">Collectio<br/>n</a> <sup>[2427]</sup> | Read only<br>Gets a collection representing the currently selected elements on the diagram.<br><br>You can remove objects from this collection to deselect them, and add elements to the collection by passing the Object ID as a name to select them. |          |
| <b>ShowDetails</b>         | <i>Long</i>                                       | Read/Write<br>A flag to indicate that the Diagram Details text should be shown: <b>1</b> = Show, <b>0</b> = Hide.                                                                                                                                      |          |
| <b>ShowPackageContents</b> | <i>Boolean</i>                                    | Read/Write<br>A flag to indicate that the package contents should be shown in the current diagram.                                                                                                                                                     |          |

| Attribute            | Type                                          | Notes                                                                                                                                    | See also                                            |
|----------------------|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
|                      |                                               |                                                                                                                                          |                                                     |
| <b>ShowPrivate</b>   | <i>Boolean</i>                                | Read/Write<br>A flag to show or hide Private features.                                                                                   |                                                     |
| <b>ShowProtected</b> | <i>Boolean</i>                                | Read/Write<br>A flag to show or hide Protected features.                                                                                 |                                                     |
| <b>ShowPublic</b>    | <i>Boolean</i>                                | Read/Write<br>A flag to show or hide Public features.                                                                                    |                                                     |
| <b>Stereotype</b>    | <i>String</i>                                 | Read/Write<br>Sets or gets the stereotype for this diagram.                                                                              |                                                     |
| <b>StyleEx</b>       | <i>String</i>                                 | Read/Write<br>Advanced style settings, reserved for the use of Sparx Systems.                                                            |                                                     |
| <b>Swimlanes</b>     | <i>String</i>                                 | Read/Write<br>Information on swimlanes contained in the diagram.<br>Please note that this property is superseded by <b>SwimlaneDef</b> . | <a href="#">SwimlaneDef Class</a> <sup>[2531]</sup> |
| <b>SwimlaneDef</b>   | <a href="#">SwimlaneDef</a> <sup>[2531]</sup> | Read/Write<br>Information on swimlanes contained in the diagram.                                                                         |                                                     |
| <b>Type</b>          | <i>String</i>                                 | Read only<br>The diagram type; see the <i>t_diagramtypes</i> table in the .EAP file for more information.                                |                                                     |
| <b>Version</b>       | <i>String</i>                                 | Read/Write<br>The version of the diagram.                                                                                                |                                                     |

**Diagram Methods:**

| Method                | Type           | Notes                                                          | See also |
|-----------------------|----------------|----------------------------------------------------------------|----------|
| <b>ApplyGroupLock</b> | <i>Boolean</i> | Applies a group lock to this diagram object, for the specified |          |

| Method                                                   | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See also                              |
|----------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|
| <b>(string aGroupName)</b>                               | <i>n</i>       | group, on behalf of the current user.<br><br>This throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information.<br><br>Parameter: <ul style="list-style-type: none"> <li>aGroupName: String - the name of the user group for which to set the group lock</li> </ul>                                                                                                                                                                                                                                                                                        |                                       |
| <b>ApplyUserLock ()</b>                                  | <i>Boolean</i> | Applies a user lock to this diagram object, for the current user.<br><br>This throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information.                                                                                                                                                                                                                                                                                                                                                                                                                |                                       |
| <b>GetLastError ()</b>                                   | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used, as an exception is thrown when an error occurs.                                                                                                                                                                                                                                                                                                                                                                                                            |                                       |
| <b>ReleaseUserLock ()</b>                                | <i>Boolean</i> | Releases a group lock or user lock on this diagram object.<br><br>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information.                                                                                                                                                                                                                                                                                                                                                                                                                            |                                       |
| <b>ReorderMessages ()</b>                                | <i>Void</i>    | Resets the display order of Sequence and Collaboration messages.<br><br>This is typically used after inserting or deleting messages in the diagram.                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                       |
| <b>ShowAsElementList (bool ShowAsList, bool Persist)</b> | <i>Boolean</i> | Toggles the diagram display between diagram format and Diagram List depending on the value of <i>ShowAsList</i> .<br><br>If <i>Persist</i> is set, the display format is written to the database so the diagram always opens in that format (diagram or list). Otherwise, the display format falls back to the default (diagram) once the display is closed.<br><br>Parameters: <ul style="list-style-type: none"> <li>ShowAsList: Boolean - indicates diagram or Diagram List</li> <li>Persist: Boolean - indicates set (maintain <i>ShowAsList</i> value) or not (revert to default)</li> </ul> | <a href="#">Diagram List</a><br>[596] |
| <b>Update ()</b>                                         | <i>Boolean</i> | Updates this diagram object after modification or appending a new item.<br><br>If <b>false</b> is returned, use <i>GetLastError()</i> to retrieve error information.                                                                                                                                                                                                                                                                                                                                                                                                                              |                                       |

| Method | Type | Notes | See also |
|--------|------|-------|----------|
|        |      |       |          |

### 19.2.2.7.2 DiagramLinks Class

A *DiagramLink* is an object that holds display information about a connector between two elements in a specific diagram. It includes, for example, the custom points and display appearance. It can be accessed from the Diagram **DiagramLinks** collection.

**Associated table in .EAP file:** *t\_diagramlinks*

#### DiagramLinks Attributes:

| Attribute          | Type                                 | Notes                                                                             |
|--------------------|--------------------------------------|-----------------------------------------------------------------------------------|
| <b>ConnectorID</b> | <i>Long</i>                          | Read/Write<br>The ID of the associated connector.                                 |
| <b>DiagramID</b>   | <i>Long</i>                          | Read/Write<br>The local ID for the associated diagram.                            |
| <b>Geometry</b>    | <i>String</i>                        | Read/Write<br>The geometry associated with the current connector in this diagram. |
| <b>InstanceID</b>  | <i>Long</i>                          | Read only<br>The connector identifier for the current model.                      |
| <b>IsHidden</b>    | <i>Boolean</i>                       | Read/Write<br>A flag to indicate if this item is hidden or not.                   |
| <b>ObjectType</b>  | <a href="#">ObjectType</a><br>[2419] | Read only<br>Distinguishes objects referenced through a Dispatch interface.       |
| <b>Path</b>        | <i>String</i>                        | Read/Write<br>The path of the connector in this diagram.                          |
| <b>Style</b>       | <i>String</i>                        | Read/Write<br>Additional style information; for example, color or thickness.      |

**DiagramLinks Methods:**

| Method                 | Type           | Notes                                                                                                                                                                                     |
|------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.     |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current <i>DiagramLink</i> object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

**Learn more**

- [DiagramLinks](#) 

**19.2.2.7.3 DiagramObject Class**

The *DiagramObject* Class stores presentation information that indicates what is displayed in a diagram and how it is shown.

**Associated table in .EAP file:** *t\_diagramobjects*

**DiagramObject Attributes:**

| Attribute         | Type        | Notes                                                                                | See also |
|-------------------|-------------|--------------------------------------------------------------------------------------|----------|
| <b>Bottom</b>     | <i>Long</i> | Read/Write<br><br>The bottom edge position of the element.                           |          |
| <b>DiagramID</b>  | <i>Long</i> | Read/Write<br><br>The ID of the associated diagram.                                  |          |
| <b>ElementID</b>  | <i>Long</i> | Read/Write<br><br>The <i>ElementID</i> of the object instance in this diagram.       |          |
| <b>InstanceID</b> | <i>Long</i> | Read<br><br>The attribute that holds the connector identifier for the current model. |          |
| <b>Left</b>       | <i>Long</i> | Read/Write                                                                           |          |

| Attribute         | Type                                                             | Notes                                                                                                                                                                           | See also                                            |
|-------------------|------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
|                   |                                                                  | The left edge position of the element.                                                                                                                                          |                                                     |
| <b>ObjectType</b> | <a href="#">ObjectTyp</a><br><a href="#">e</a> <sup>[2419]</sup> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                                                     |                                                     |
| <b>Right</b>      | <i>Long</i>                                                      | Read/Write<br>The right edge position of the element.                                                                                                                           |                                                     |
| <b>Sequence</b>   | <i>Long</i>                                                      | Read/Write<br>The sequence position when loading the object into the diagram (this affects its Z order).<br>The Z-order is one-based and the lowest value is in the foreground. |                                                     |
| <b>Style</b>      | <i>Variant</i>                                                   | Write only (reading this value gives undefined results)<br>The style information for this object.                                                                               | <a href="#">Setting the Style</a> <sup>[2531]</sup> |
| <b>Top</b>        | <i>Long</i>                                                      | Read/Write<br>The top edge position of the element.                                                                                                                             |                                                     |

### DiagramObject Methods

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current <i>DiagramObject</i> after modification or appending a new item<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.   |

### Topics

| Topic              | Detail                                                                                       | See also |
|--------------------|----------------------------------------------------------------------------------------------|----------|
| <b>Setting The</b> | The <i>Style</i> attribute is used for setting the appearance of a <i>DiagramObject</i> ; it |          |

| Topic        | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Style</b> | <p>is set with a string value in the format:</p> <p><code>BCol =n; BFol =n; LCol =n; LWth =n;</code></p> <p>where:</p> <ul style="list-style-type: none"> <li>• <i>BCol</i> = Background Color</li> <li>• <i>BFol</i> = Font Color</li> <li>• <i>LCol</i> = Line Color</li> <li>• <i>LWth</i> = Line Width</li> </ul> <p>The color value is a decimal representation of the hex RGB value, where Red=FF, Green=FF00 and Blue=FF0000</p> <p><code>DiagramStyle = " BCol =35723; BFol =9342520; LCol =9342520; LWth =1; "</code></p> <p>The following code snippet shows how you might change the style settings for all of the objects in the current diagram, in this case changing everything to red:</p> <pre> For Each aDiagramObj In aDiagram.DiagramObjects     aDiagramObj.Style = " BCol =255; BFol =9342520; LCol =9342520; LWth =1; "     aDiagramObj.Update aRepos.ReloadDiagram aDiagramObj . Diagram.D Next </pre> |          |

#### 19.2.2.7.4 SwimlaneDef Class

A *SwimlaneDef* object makes available attributes relating to a single row or column in a list of swimlanes.

| Attribute             | Type           | Notes                                                        |
|-----------------------|----------------|--------------------------------------------------------------|
| <b>Bold</b>           | <i>Boolean</i> | Read/Write<br>Show the title text in bold.                   |
| <b>FontColor</b>      | <i>Long</i>    | Read/Write<br>The RGB color used to draw the titles.         |
| <b>HideClassifier</b> | <i>Boolean</i> | Read/Write<br>Removes any classifier from the title display. |
| <b>HideNames</b>      | <i>Boolean</i> | Read/Write                                                   |



| Attribute             | Type                                                | Notes                                                                                                              |
|-----------------------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
|                       |                                                     | Set to <b>true</b> to hide the swimlane titles.                                                                    |
| <b>LineColor</b>      | <i>Long</i>                                         | Read/Write<br>The RGB color used to draw swimlane borders.                                                         |
| <b>LineWidth</b>      | <i>Long</i>                                         | Read/Write<br>The width of the line, in pixels, used to draw swimlanes. Valid values are <b>1, 2</b> or <b>3</b> . |
| <b>Locked</b>         | <i>Boolean</i>                                      | Read/Write<br>If set to <b>true</b> , disables user modification of the swimlanes via the diagram.                 |
| <b>ObjectType</b>     | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                        |
| <b>Orientation</b>    | <i>String</i>                                       | Read/Write<br>Indicates whether the swimlanes are vertical or horizontal.                                          |
| <b>ShowInTitleBar</b> | <i>Boolean</i>                                      | Read/Write<br>Enables vertical swimlane titles to be shown in the title bar.                                       |
| <b>Swimlanes</b>      | <a href="#">Swimlanes</a><br><small>[2532]</small>  | Read/Write<br>A list of individual swimlanes.                                                                      |

#### 19.2.2.7.5 Swimlanes Class

A *Swimlanes* object is attached to a diagram's *SwimlaneDef* object and provides a mechanism to access individual swimlanes.

##### Swimlanes Attributes:

| Attribute         | Type                                                | Notes                                        |
|-------------------|-----------------------------------------------------|----------------------------------------------|
| <b>Count</b>      | <i>Long</i>                                         | Read/Write<br>Gives the number of swimlanes. |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[2419]</small> | Read only                                    |

| Attribute | Type | Notes                                                          |
|-----------|------|----------------------------------------------------------------|
|           |      | Distinguishes objects referenced through a Dispatch interface. |

**Swimlanes Methods:**

| Method                                               | Type                                                                   | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------------------|------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Add (string Title, long Width)</b>                | <a href="#">Swimlane</a><br><small>[2534]</small>                      | <p>Adds a new swimlane to the end of the list, and returns a swimlane object representing the newly added entry.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Title: String - The title text that appears at the top of the swimlane; this can be the same as an existing swimlane title</li> <li>Width: Long - The width of the swimlane in pixels</li> </ul>                                                                                                              |
| <b>Delete (object Index)</b>                         | <i>Void</i>                                                            | <p>Deletes a selected swimlane.</p> <p>If the string matches more than one entry, only the first entry is deleted.</p> <p>Parameter:</p> <ul style="list-style-type: none"> <li>Index: Object - Either a string representing the title text or an integer representing the zero-based index of the swimlane to delete</li> </ul>                                                                                                                                                             |
| <b>DeleteAll ()</b>                                  | <i>Void</i>                                                            | Removes all swimlanes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Insert (long Index, string Title, long Width)</b> | <a href="#">Swimlane</a><br><small>[2534]</small>                      | <p>Inserts a swimlane at a specific position, and returns a swimlane object representing the newly added entry.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Index: Long - The zero-based index of the existing Swimlane before which this new entry is inserted</li> <li>Title: String - The title text which appears at the top of the swimlane; this can be the same as an existing swimlane title</li> <li>Width: Long - The width of the swimlane in pixels</li> </ul> |
| <b>Items (object Index)</b>                          | <a href="#">Swimlane</a><br><small>[2534]</small><br><i>collection</i> | <p>Accesses an individual swimlane.</p> <p>If the string matches more than one swimlane title, the first matching swimlane is returned.</p> <p>Parameter:</p> <ul style="list-style-type: none"> <li>Index: Object - Either a string representing the title text or an integer representing the zero-based index of the swimlane to get</li> </ul>                                                                                                                                           |

| Method | Type | Notes |
|--------|------|-------|
|        |      |       |

#### Learn more

- [SwimlaneDef Class](#) [2531]

#### 19.2.2.7.6 *Swimlane Class*

A *Swimlane* object makes available attributes relating to a single row or column in a list of swimlanes.

| Attribute             | Type                                             | Notes                                                                                                                                           |
|-----------------------|--------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>BackColor</b>      | <i>Long</i>                                      | Read/Write<br>The RGB color that the swimlane is filled with.                                                                                   |
| <b>ClassifiedGuid</b> | <i>String</i>                                    | Read/Write<br>The GUID of the classifier Class. This can be obtained from the corresponding Element object via the <i>ElementGUID</i> property. |
| <b>ObjectType</b>     | <a href="#">ObjectType</a> <small>[2413]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface.                                                                     |
| <b>Title</b>          | <i>String</i>                                    | Read/Write<br>The text at the head of the swimlane.                                                                                             |
| <b>Width</b>          | <i>Long</i>                                      | Read/Write<br>The width of the swimlane, in pixels.                                                                                             |

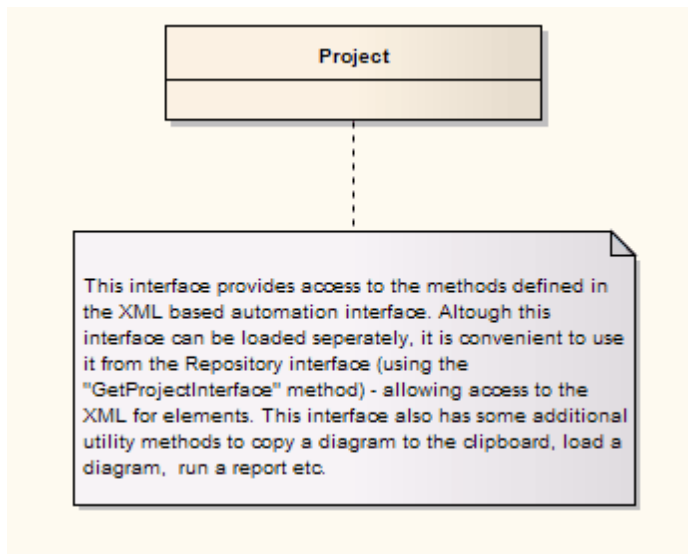
#### Learn more

- [Swimlane Class](#) [2532]

#### 19.2.2.8 *Project Interface Package*

The *Enterprise Architect.Project* interface. This is the XML-based interface to Enterprise Architect elements; it also includes some utility functions. You can get a pointer to this interface using the *Repository*. *GetProjectInterface* method.

#### Example



### 19.2.2.8.1 Project Class

The Project interface can be accessed from the Repository using *GetProjectInterface()*. The returned interface provides access to the XML-based Enterprise Architect Automation Interface. Use this interface to get XML for the various internal elements and to run some utility functions to perform tasks such as load diagrams or run reports.

#### Project Attributes:

| Attribute         | Type                                         | Notes                                                                       |
|-------------------|----------------------------------------------|-----------------------------------------------------------------------------|
| <b>ObjectType</b> | <a href="#">ObjectType</a> <sup>[2419]</sup> | Read only<br>Distinguishes objects referenced through a Dispatch interface. |

#### Project Methods:

| Method                                                                   | Type           | Notes                                                                                                                                                                                 | See also                                           |
|--------------------------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| <b>CancelValidation ()</b>                                               | <i>Void</i>    | Cancels a validation process.                                                                                                                                                         | <a href="#">Model Validation</a> <sup>[2203]</sup> |
| <b>CanValidate ()</b>                                                    | <i>Boolean</i> | Returns a value to indicate that the Model Validation component is loaded.                                                                                                            |                                                    |
| <b>CreateBaseline (string PackageGUID, string Version, string Notes)</b> | <i>Boolean</i> | Creates a Baseline of a specified package.<br>Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to Baseline</li> </ul> |                                                    |

| Method                                                                                                 | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See also                                                                                                                                                    |
|--------------------------------------------------------------------------------------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                        |                | <ul style="list-style-type: none"> <li>Version: String - the version of the Baseline</li> <li>Notes: String - any notes concerning the Baseline</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                             |
| <b>CreateBaselineEx(string PackageGUID, string Version, string Notes, EA.CreateBaselineFlag Flags)</b> | <i>Boolean</i> | <p>Creates a Baseline of a specified package, with a flag to <b>exclude package contents</b> below the first level.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to be Baselined</li> <li>Version: String - the version of the Baseline</li> <li>Notes: String - any notes concerning the Baseline</li> <li>Flags: <i>EA.CreateBaselineFlag</i> - whether or not to exclude the package contents below the first level</li> </ul>                                                                                                                                                                                                                                                                                                                                | <a href="#">Create Baselines</a><br><a href="#">EA</a><br><a href="#">CreateBaselineFlag</a>                                                                |
| <b>DefineRule (string CategoryID, EA.EnumMVErrorType ErrorType, string ErrorMessage)</b>               | <i>String</i>  | <p>Defines the individual rules that can be performed during model validation. It must be called once for each rule from the <i>EA_OnInitializeUserRules</i> broadcast handler.</p> <p>The return value is a <i>RuleId</i>, which can be used for reference purposes when an individual rule is executed by Enterprise Architect during model validation.</p> <p>See the <i>Model Validation Example</i> for a detailed example of the use of this method.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>CategoryId: String - should be passed the return value from the <i>DefineRuleCategory</i> method</li> <li>ErrorType: EA.EnumMVErrorType - depending on the severity of the error being validated, can be: <ul style="list-style-type: none"> <li><b>mvErrorCritical</b></li> <li><b>mvError</b></li> </ul> </li> </ul> | <a href="#">EA_OnInitializeUserRules</a><br><a href="#">Model Validation Example</a><br><a href="#">DefineRuleCategory</a><br><a href="#">PublishResult</a> |

| Method                                                                                  | Type    | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See also                                                                                                                                                                 |
|-----------------------------------------------------------------------------------------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                         |         | <ul style="list-style-type: none"> <li>• <b>mvWarning</b>, or</li> <li>• <b>mvInformation</b></li> </ul> <ul style="list-style-type: none"> <li>• ErrorMessage: String - can contain a default error string, although this is probably overridden by the <i>PublishResult</i> call</li> </ul>                                                                                                                                                                                                                                                                                            |                                                                                                                                                                          |
| <b>DefineRuleCategory</b><br>(string CategoryName)                                      | String  | <p>Defines a category of rules that can be performed during model validation (there is typically one category per Add-In). It must be called once from the <i>EA_OnInitializeUserRules</i> broadcast handler.</p> <p>The return value is a CategoryId that must be passed to the <i>DefineRule</i> method.</p> <p>See the <i>Model Validation Example</i> for a detailed example of the use of this method.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• CategoryName: String - a text string that is visible in the Model Validation Configuration dialog</li> </ul> | <a href="#">EA_OnInitializeUserRules</a> <sup>[2622]</sup><br><a href="#">DefineRule</a> <sup>[2536]</sup><br><a href="#">Model Validation Example</a> <sup>[2630]</sup> |
| <b>DeleteBaseline</b> (string BaselineGUID)                                             | Boolean | <p>Deletes a Baseline, identified by the BaselineGUID, from the repository.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• BaselineGUID: String - the GUID (in XML format) of the Baseline to delete</li> </ul>                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                          |
| <b>DoBaselineCompare</b><br>(string PackageGUID, string Baseline, string ConnectString) | String  | <p>Performs a Baseline comparison using the supplied package GUID and Baseline GUID (obtained in the result list from <i>GetBaselines</i>).</p> <p>Optionally you can include the connection string required to find the Baseline if it exists in a different model file.</p> <p>This method returns a log file of the status of all elements found and compared in the difference procedure. You can use this log information as input to <i>DoBaselineMerge</i> - automatically merging information from the Baseline.</p>                                                             | <a href="#">GetBaselines</a> <sup>[2546]</sup><br><a href="#">DoBaselineMerge</a> <sup>[2538]</sup>                                                                      |

| Method                                                                                                                  | Type          | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also                                                    |
|-------------------------------------------------------------------------------------------------------------------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
|                                                                                                                         |               | <p>Parameters:</p> <ul style="list-style-type: none"> <li>• PackageGUID: String - the GUID (in XML format) of the package to run the comparison on</li> <li>• Baseline: String - the GUID (in XML format) of the Baseline to run the comparison on</li> <li>• ConnectString: String - the location of the external .EAP file or DBMS to extract the Baseline from</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                             |
| <p><b>DoBaselineMerge</b><br/>(string PackageGUID, string Baseline, string MergeInstructions, string ConnectString)</p> | <p>String</p> | <p>Performs a batch merge based on instructions contained in an XML file (<i>MergeInstructions</i>). You can supply an optional connection string if the Baseline is located in another model.</p> <p>In the <i>MergeInstructions</i> file, each <i>MergeItem</i> node supplies the GUID of a differenced item from the XML difference log. As the merge is uni-directional and actioned in only one possible way, no additional arguments are required. Enterprise Architect chooses the correct procedure based on the Difference results.</p> <pre>&lt;Merge&gt;   &lt;MergeItem guid="{XXXXXXX}" /&gt;   &lt;MergeItem guid="{XXXXXXX}" /&gt; &lt;/Merge&gt;</pre> <p>Alternatively, you can supply a single <i>MergeItem</i> with a GUID of <i>RestoreAll</i>. In this case, Enterprise Architect batch-processes ALL differences.</p> <pre>&lt;Merge&gt;   &lt;MergeItem guid="RestoreAll"   changed="true"   baselineOnly="true"   modelOnly="true"   moved="true" fullRestore="false" /&gt; &gt; &lt;/Merge&gt;</pre> | <p><a href="#">DoBaselineCompare()</a><sup>[2537]</sup></p> |

| Method                                          | Type                                 | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See also |
|-------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                 |                                      | <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to merge the Baseline into</li> <li>Baseline: String - the GUID of the Baseline (in XML format) to merge into the package</li> <li>MergeInstructions: String - the file containing the GUID of each differenced item from the XML difference log returned by <i>DoBaselineCompare()</i></li> <li>ConnectionString: String - the location of the EAP file or DBMS to get the Baseline from, if not in the same model as the package</li> </ul> |          |
| <b>EnumDiagramElements (string DiagramGUID)</b> | protected abstract:<br><i>String</i> | <p>Gets an XML list of all elements in a diagram.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to get elements for</li> </ul>                                                                                                                                                                                                                                                                                                                                                          |          |
| <b>EnumDiagrams (string PackageGUID)</b>        | protected abstract:<br><i>String</i> | <p>Gets an XML list of all diagrams in a specified package.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to list diagrams for</li> </ul>                                                                                                                                                                                                                                                                                                                                               |          |
| <b>EnumElements (string PackageGUID)</b>        | protected abstract:<br><i>String</i> | <p>Gets an XML list of elements in a specified package.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to get a list of elements for</li> </ul>                                                                                                                                                                                                                                                                                                                                          |          |



| Method                                                                                                                                                | Type                                 | Notes                                                                                                                                                                                                                                                       | See also                                        |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| <b>EnumLinks (string ElementGUID)</b>                                                                                                                 | protected abstract:<br><i>String</i> | Gets an XML list of connectors for a specified element.<br><br>Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element to get all associated connectors for</li> </ul>                             |                                                 |
| <b>EnumPackages (string PackageGUID)</b>                                                                                                              | protected abstract:<br><i>String</i> | Gets an XML list of child packages inside a parent package.<br><br>Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the parent package</li> </ul>                                                       |                                                 |
| <b>EnumProjects ()</b>                                                                                                                                | protected abstract:<br><i>String</i> | Gets a list of projects in the current file; corresponds to <i>Models</i> in <i>Repository</i> .                                                                                                                                                            | <a href="#">Models</a> <sup>[2450]</sup>        |
| <b>EnumViews ()</b>                                                                                                                                   | protected abstract:<br><i>String</i> | Enumerates the Views for a project. Returned as an XML document.                                                                                                                                                                                            |                                                 |
| <b>EnumViewEx (string ProjectGUID)</b>                                                                                                                | protected abstract:<br><i>String</i> | Gets a list of Views in the current project.<br><br>Parameters: <ul style="list-style-type: none"> <li>ProjectGUID: String - the GUID (in XML format) of the project to get views for</li> </ul>                                                            |                                                 |
| <b>Exit ()</b>                                                                                                                                        | protected abstract:<br><i>String</i> | Exits the current instance of Enterprise Architect; this function is maintained for backward compatibility and should never be called.<br><br>Enterprise Architect automatically exits when you are no longer using any of the provided objects.            |                                                 |
| <b>ExportPackageXML (string PackageGUID, enumXMIMType XMIMType, long DiagramXML, long DiagramImage, long FormatXML, long UseDTD, string FileName)</b> | protected abstract:<br><i>String</i> | Exports XMI for a specified package.<br><br>Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to be exported</li> <li>XMIMType: EnumXMIMType - specifies the XMI type and version</li> </ul> | <a href="#">XMIMType Enum</a> <sup>[2423]</sup> |

| Method                                                                                                                                                                                    | Type                                 | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See also                                                                                                                                                      |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                                           |                                      | <p>information; see <i>XMIType Enum</i> for accepted values</p> <ul style="list-style-type: none"> <li>DiagramXML: Long - <b>true</b> if XML for diagrams is required; accepted values: <ul style="list-style-type: none"> <li><b>0</b> = Do not export diagrams</li> <li><b>1</b> = Export diagrams</li> <li><b>2</b> = Export diagrams along with alternate images</li> </ul> </li> <li>DiagramImage: Long - the format for diagram images to be created at the same time; accepted values: <ul style="list-style-type: none"> <li><b>-1</b>=NONE</li> <li><b>0</b>=EMF</li> <li><b>1</b>=BMP</li> <li><b>2</b>=GIF</li> <li><b>3</b>=PNG</li> <li><b>4</b>=JPG</li> </ul> </li> <li>FormatXML: Long - <b>true</b> if XML output should be formatted prior to saving</li> <li>UseDTD: Long - <b>true</b> if a DTD should be used</li> <li>FileName: String - the filename to output to</li> </ul> |                                                                                                                                                               |
| <b>ExportPackageXMILEx</b><br>(string PackageGUID, enumXMIType XMIType, long DiagramXML, long DiagramImage, long FormatXML, long UseDTD, string FileName, ea. ExportPackageXMIFlag Flags) | protected abstract:<br><i>String</i> | Exports XML for a specified package, with a flag to determine whether the export <b>includes package content</b> below the first level.<br>Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to be exported</li> <li>XMIType: EnumXMIType - specifies</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <a href="#">Package Content</a> <sup>[426]</sup><br><a href="#">XMIType Enum</a> <sup>[2423]</sup><br><a href="#">ExportPackageXML Flag</a> <sup>[2418]</sup> |

| Method                                                         | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See also |
|----------------------------------------------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                                |                | <p>the XML type and version information; see <i>XMType Enum</i> for accepted values</p> <ul style="list-style-type: none"> <li>DiagramXML: Long - <b>true</b> if XML for diagrams is required; accepted values: <ul style="list-style-type: none"> <li><b>0</b> = Do not export diagrams</li> <li><b>1</b> = Export diagrams</li> <li><b>2</b> = Export diagrams along with alternate images</li> </ul> </li> <li>DiagramImage: Long - the format for diagram images to be created at the same time; accepted values: <ul style="list-style-type: none"> <li><b>-1</b>=NONE</li> <li><b>0</b>=EMF</li> <li><b>1</b>=BMP</li> <li><b>2</b>=GIF</li> <li><b>3</b>=PNG</li> <li><b>4</b>=JPG</li> </ul> </li> <li>FormatXML: Long - <b>true</b> if XML output should be formatted prior to saving</li> <li>UseDTD: Long - <b>true</b> if a DTD should be used.</li> <li>FileName: String - the filename to output to</li> <li>Flags: ea.ExportPackageXMIFlag - whether or not to include package content below the first level (currently only supported for <i>xmiEADefault</i>)</li> </ul> |          |
| <b>GenerateClass (string ElementGUID, string ExtraOptions)</b> | <i>Boolean</i> | Generates the code for a single Class.<br>Parameters:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |          |

| Method                                                                                                                      | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See also                                                   |
|-----------------------------------------------------------------------------------------------------------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
|                                                                                                                             |                | <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element to generate</li> <li>ExtraOptions: String - enables extra options to be given to the command; currently unused</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                            |
| <b>GenerateDiagramFromScenario (string ElementGUID, EnumScenarioDiagramType DiagramType, long OverwriteExistingDiagram)</b> | <i>Boolean</i> | <p>Generates various diagrams from the Structured Specification of an element.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element containing the Structured Specification</li> <li>DiagramType: EnumScenarioDiagramType - the type of diagram to generate; see <i>ScenarioDiagramType Enum</i> for accepted values</li> <li>OverwriteExistingDiagram: Long - determines whether to overwrite the existing diagram or synchronize the existing elements with the scenario steps</li> </ul> <p><b>0</b> = Delete the existing diagram and elements, and create a new diagram and elements</p> <p><b>1</b> = Synchronize existing elements with the scenario steps and preserve the diagram layout</p> <p><b>2</b> = Synchronize existing elements with the scenario steps and re-cast the diagram layout</p> <p><b>3</b> = Do not generate a diagram if one already exists</p> | <a href="#">ScenarioDiagramType Enum</a> <sup>[2422]</sup> |
| <b>GenerateElementDDL (string ElementGUID, string</b>                                                                       | <i>Boolean</i> | Generates DDL for an element.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                            |

| Method                                                                   | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | See also |
|--------------------------------------------------------------------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>FileName, string<br/>ExtraOptions)</b>                                |                | <p>For example:</p> <pre>owner = true; pkfkconstraints = true; commentlevel = 3;</pre> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• ElementGUID: String – the GUID (in XML format) of the element to generate DDL for</li> <li>• FileName: String – the target file path to which to write the generated DDL</li> <li>• ExtraOptions: String – enables extra options to be given to the command; currently enables: <ul style="list-style-type: none"> <li>• Generate Owner (<i>owner</i>)</li> <li>• Generate PK/FK Key Constraints (<i>pkfkconstraints</i>)</li> <li>• Generate Comments: <pre>0=None 1=Table 2=Column 3=All</pre> </li> </ul> </li> </ul> |          |
| <b>GeneratePackage (string<br/>PackageGUID, string<br/>ExtraOptions)</b> | <i>Boolean</i> | <p>Generates the code for all Classes within a package.</p> <p>For example:</p> <pre>recurse=1; overwrite=1; dir=C:\</pre> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• PackageGUID: String - the GUID (in XML format) of the package to generate code for</li> <li>• ExtraOptions: String - enables extra options to be given to the command; currently enables: <ul style="list-style-type: none"> <li>• Generation of all subpackages (</li> </ul> </li> </ul>                                                                                                                                                                                            |          |

| Method                                                                               | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See also |
|--------------------------------------------------------------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                                                      |                | <p><i>recurse</i>)</p> <ul style="list-style-type: none"> <li>Force overwrite of all files (<i>overwrite</i>) and</li> <li>Specification to auto generate all paths (<i>dir</i>)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |          |
| <b>GeneratePackageDDL</b> (string PackageGUID, string FileName, string ExtraOptions) | <i>Boolean</i> | <p>Generates DDL for a package.</p> <p>For example:</p> <pre>owner=true; pkfkconstraints=true; commentlevel=3; singlefile=true; includechild=true;</pre> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String – the GUID (in XML format) of the package to generate DDL for</li> <li>FileName: String – the target file path to which to write the generated DDL</li> <li>ExtraOptions: String – enables extra options to be given to the command; currently enables: <ul style="list-style-type: none"> <li>Generate DDL for child packages (<i>includechild</i>)</li> <li>Generate Owner (<i>owner</i>)</li> <li>Generate PK/FK Key Constraints (<i>pkfkconstraints</i>)</li> <li>Generate Comments: <ul style="list-style-type: none"> <li>0 - None</li> <li>1 - Table</li> <li>2 - Column</li> <li>3 - All</li> </ul> </li> </ul> </li> <li>Generate in SingleFile (<i>singlefile</i>)</li> </ul> |          |

| Method                                                                                    | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See also                                                                                                                   |
|-------------------------------------------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>GenerateTestFromScenario (string ElementGUID, EnumScenarioTestType TestType)</b>       | <i>Boolean</i> | <p>Generates either an <b>Internal test</b> or an <b>External test</b> from the Structured Specification of an element.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element containing the Structured Specification</li> <li>TestType: EnumScenarioTestType - the type of test to generate; see <i>ScenarioTestType Enum</i> for accepted values</li> </ul>                                                                                                                                                                                                                                             | <p><a href="#">Generate Test Cases</a> <sup>[873]</sup></p> <p><a href="#">ScenarioTestType Enum</a> <sup>[2423]</sup></p> |
| <b>GenerateXSD (string PackageGUID, string FileName, string Encoding, string Options)</b> | <i>Boolean</i> | <p>Creates an XML schema for this <i>GenerateXSD</i>. Returns <b>true</b> on success.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package</li> <li>FileName: String - the target filepath</li> <li>Encoding: String - the XML encoding for the code page instruction</li> <li>Options: String - enables extra options to be given to the command; currently enables: <ul style="list-style-type: none"> <li><i>GenGlobalElement</i> - turn the generation of global elements for all global <i>ComplexTypes</i> <b>On</b> or <b>Off</b>; for example: -<br/>GenGlobalElement = 1</li> </ul> </li> </ul> |                                                                                                                            |
| <b>GetBaselines (string PackageGUID, string ConnectString)</b>                            | <i>String</i>  | <p>Returns a list (in XML format) of Baselines associated with the supplied package GUID. Optionally, you can provide a connection string to get Baselines from the same package, but located in a different model file (or DBMS).</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to get</li> </ul>                                                                                                                                                                                                                                                                                                |                                                                                                                            |

| Method                                            | Type                                 | Notes                                                                                                                                                                                             | See also |
|---------------------------------------------------|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                   |                                      | Baselines for <ul style="list-style-type: none"> <li>• ConnectString: String - the location of the EAP file or DBMS to get the Baselines from, if not in the same model as the package</li> </ul> |          |
| <b>GetDiagram (string DiagramGUID)</b>            | protected abstract:<br><i>String</i> | Gets the diagram details, in XML format.<br>Parameters: <ul style="list-style-type: none"> <li>• DiagramGUID: String - the GUID (in XML format) of the diagram to get details for</li> </ul>      |          |
| <b>GetElement (string ElementGUID)</b>            | protected abstract:<br><i>String</i> | Gets XML for the specified element.<br>Parameters: <ul style="list-style-type: none"> <li>• ElementGUID: String - the GUID (in XML format) of the element to retrieve XML for</li> </ul>          |          |
| <b>GetElementConstraints (string ElementGUID)</b> | protected abstract:<br><i>String</i> | Gets constraints for an element, in XML format.<br>Parameters: <ul style="list-style-type: none"> <li>• ElementGUID: String - the GUID (in XML format) of the element</li> </ul>                  |          |
| <b>GetElementEffort (string ElementGUID)</b>      | protected abstract:<br><i>String</i> | Gets efforts for an element, in XML format.<br>Parameters: <ul style="list-style-type: none"> <li>• ElementGUID: String - the GUID (in XML format) of the element</li> </ul>                      |          |
| <b>GetElementFiles (string ElementGUID)</b>       | protected abstract:<br><i>String</i> | Gets metrics for an element, in XML format.<br>Parameters: <ul style="list-style-type: none"> <li>• ElementGUID: String - the GUID (in XML format) of the element</li> </ul>                      |          |
| <b>GetElementMetrics (string ElementGUID)</b>     | protected abstract:<br><i>String</i> | Gets files for an element, in XML format.<br>Parameters: <ul style="list-style-type: none"> <li>• ElementGUID: String - the GUID (in XML format) of the element</li> </ul>                        |          |




| Method                                             | Type                                 | Notes                                                                                                                                                                                                         | See also |
|----------------------------------------------------|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                    |                                      | XML format) of the element                                                                                                                                                                                    |          |
| <b>GetElementProblems (string ElementGUID)</b>     | protected abstract:<br><i>String</i> | Gets a list of issues (problems) associated with an element, in XML format.<br>Parameters:<br><ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul> |          |
| <b>GetElementProperties (string ElementGUID)</b>   | protected abstract:<br><i>String</i> | Gets Tagged values for an element, in XML format.<br>Parameters:<br><ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>                           |          |
| <b>GetElementRequirements (string ElementGUID)</b> | protected abstract:<br><i>String</i> | Gets a list of requirements for an element, in XML format.<br>Parameters:<br><ul style="list-style-type: none"> <li>ElementGUID: String -the GUID (in XML format) of the element</li> </ul>                   |          |
| <b>GetElementResources (string ElementGUID)</b>    | protected abstract:<br><i>String</i> | Gets a list of resources for an element, in XML format.<br>Parameters:<br><ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>                     |          |
| <b>GetElementRisks (string ElementGUID)</b>        | protected abstract:<br><i>String</i> | Gets a list of risks associated with an element, in XML format.<br>Parameters:<br><ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>             |          |
| <b>GetElementScenarios (string ElementGUID)</b>    | protected abstract:<br><i>String</i> | Gets a list of scenarios for an element, in XML format.<br>Parameters:<br><ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>                     |          |
| <b>GetElementTests (string ElementGUID)</b>        | protected                            | Gets a list of tests for an element, in XML                                                                                                                                                                   |          |

| Method                                                                                                                                  | Type                                 | Notes                                                                                                                                                                                                                                                                                      | See also |
|-----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>ElementGUID)</b>                                                                                                                     | abstract:<br><i>String</i>           | format.<br>Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>                                                                                                                                                     |          |
| <b>GetFileNameDialog (string Filename, string FilterString, long FilterIndex, long Flags, string InitialDirectory, long OpenOrSave)</b> | <i>String</i>                        | Opens the Standard File Open or Save dialog and returns a string on success.                                                                                                                                                                                                               |          |
| <b>GetLastError ()</b>                                                                                                                  | protected abstract:<br><i>String</i> | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.                                                                                                      |          |
| <b>GetLink (string LinkGUID)</b>                                                                                                        | protected abstract:<br><i>String</i> | Gets connector details, in XML format.<br>Parameters: <ul style="list-style-type: none"> <li>LinkGUID: String - the GUID (in XML format) of the connector to get details of</li> </ul>                                                                                                     |          |
| <b>GUIDtoXML (string GUID)</b>                                                                                                          | <i>String</i>                        | Changes an internal GUID to the form used in XML.<br>Parameters: <ul style="list-style-type: none"> <li>GUID: String - the Enterprise Architect style GUID to convert to XML format</li> </ul>                                                                                             |          |
| <b>ImportDirectory (string PackageGUID, string Language, string DirectoryPath, string ExtraOptions)</b>                                 | <i>Boolean</i>                       | Imports a source code directory into the model.<br>Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to reverse engineer code into</li> <li>Language: String - specifies the language of the code to be imported</li> </ul> |          |

| Method                                                                                             | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See also |
|----------------------------------------------------------------------------------------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                                                                    |                | <ul style="list-style-type: none"> <li>DirectoryPath: String - specifies the path where the code is found on the computer</li> <li>ExtraOptions: String - enables extra options to be given to the command; currently enables import of source from all child directories ( <i>recurse</i>) - for example: <i>recurse=1</i></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                |          |
| <b>ImportFile (string PackageGUID, string Language, string FileName, string ExtraOptions)</b>      | <i>Boolean</i> | <p>Imports an individual file or binary module into the model, in a package per namespace style import.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to reverse engineer code into; this is expected to be a namespace root package.</li> <li>Language: String - specifies the language of the code to be imported</li> </ul> <p>Use the value "<b>DNPE</b>" to import a binary module; this imports a .Net assembly or Java .class file, but <b>not</b> a .jar file.</p> <ul style="list-style-type: none"> <li>Filename: String - specifies the path where the code or module is found on the computer</li> <li>ExtraOptions: String - enables extra options to be given to the command; currently unused</li> </ul> |          |
| <b>ImportPackageXML (string PackageGUID, string Filename, long ImportDiagrams, long StripGUID)</b> | <i>String</i>  | <p>Imports an XML file at a point in the tree.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the target package to import the XML file into (or overwrite with the XML file)</li> <li>Filename or XMLText: String - the</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |          |

| Method                                                                                                                                       | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also                                                     |
|----------------------------------------------------------------------------------------------------------------------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
|                                                                                                                                              |                | <p>name of the XML file</p> <p>If the String is of type <i>filename</i> it is interpreted as a source file, otherwise the String is imported as XML text.</p> <ul style="list-style-type: none"> <li>• ImportDiagrams: Long</li> <li>• StripGUID: Long - boolean value to indicate whether to replace the element UniqueIDs on import; if stripped, then a package could be imported twice into Enterprise Architect, as two different versions</li> </ul>                                                                                                                                                         |                                                              |
| <b>LayoutDiagram</b> (string DiagramGUID, long LayoutStyle)                                                                                  | <i>Boolean</i> | <p><b>Deprecated.</b> it is recommended that <i>LayoutDiagramEx</i> is used instead.</p> <p>Calls the function to automatically layout a diagram in hierarchical fashion. It is only recommended for Class and Object diagrams.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• DiagramGUID: String - the GUID (in XML format) of the diagram to lay out</li> <li>• LayoutStyle: Long - always ignored</li> </ul>                                                                                                                                                                                  |                                                              |
| <b>LayoutDiagramEx</b> (string DiagramGUID, long LayoutStyle, long Iterations, long LayerSpacing, long ColumnSpacing, boolean SaveToDiagram) | <i>Boolean</i> | <p>Calls the function to automatically layout a diagram in hierarchical fashion. It is only recommended for Class and Object diagrams.</p> <p><i>LayoutStyle</i> accepts the following options</p> <ul style="list-style-type: none"> <li>• Default Options: <ul style="list-style-type: none"> <li>IsDiagramDefault</li> <li>IsProgramDefault</li> </ul> </li> <li>• Cycle Removal Options: <ul style="list-style-type: none"> <li>IsCycleRemoveGreedy</li> <li>IsCycleRemoveDFS</li> </ul> </li> <li>• Layering Options: <ul style="list-style-type: none"> <li>IsLayeringLongestPathSink</li> </ul> </li> </ul> | <a href="#">ConstLayoutStyles Enum</a> <small>[2414]</small> |


| Method | Type | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See also |
|--------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|        |      | <p>IsLayeringLongestPathSource</p> <p>IsLayeringOptimalLinkLength</p> <ul style="list-style-type: none"> <li>• Initialize Options: <ul style="list-style-type: none"> <li>IsInitializeNaive</li> <li>IsInitializeDFSOut</li> <li>IsInitializeDFSIn</li> </ul> </li> <li>• Crossing Reduction Option: <ul style="list-style-type: none"> <li>IsCrossReduceAggressive</li> </ul> </li> <li>• Layout Options - Direction <ul style="list-style-type: none"> <li>IsLayoutDirectionUp</li> <li>IsLayoutDirectionDown</li> <li>IsLayoutDirectionLeft</li> <li>IsLayoutDirectionRight</li> </ul> </li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• DiagramGUID: String - the GUID (in XML format) of the diagram to lay out</li> <li>• LayoutStyle: Long - the layout style</li> <li>• Iterations: Long - the number of layout iterations the Layout process should take to perform cross reduction (Default value = <b>4</b>)</li> <li>• LayerSpacing: Long - the per-element layer spacing the Layout process shall use (Default value = <b>20</b>)</li> <li>• ColumnSpacing: Long - the per-element column spacing the Layout process shall use (Default value = <b>20</b>)</li> <li>• SaveToDiagram: Boolean - specifies whether or not Enterprise Architect should save the supplied</li> </ul> |          |

| Method                                                                  | Type                                  | Notes                                                                                                                                                                                                                                                                                                                                        | See also                                                                                                      |
|-------------------------------------------------------------------------|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
|                                                                         |                                       | layout options as default to the diagram in question                                                                                                                                                                                                                                                                                         |                                                                                                               |
| <b>LoadControlledPackage (string PackageGUID)</b>                       | <i>String</i>                         | Loads a package that has been marked and configured as controlled. The filename details are stored in the package control data.<br><br>Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to load</li> </ul>                                                                   |                                                                                                               |
| <b>LoadDiagram (string DiagramGUID)</b>                                 | protected abstract:<br><i>Boolean</i> | Loads a diagram by its GUID.<br><br>Parameter: <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to load; if you retrieve the GUID using the Diagram interface, use the <i>GUIDtoXML</i> function to convert it to XML format</li> </ul>                                                  | <a href="#">GUIDtoXML</a>  |
| <b>LoadProject (string FileName)</b>                                    | protected abstract:<br><i>Boolean</i> | Loads an Enterprise Architect project file.<br><br>Do not use this method if you have accessed the Project interface from the Repository, which has already loaded a file.<br><br>Parameters: <ul style="list-style-type: none"> <li>FileName: String - the name of the project file to load</li> </ul>                                      |                                                                                                               |
| <b>Migrate (string GUID, string SourceType, string DestinationType)</b> | <i>Void</i>                           | Migrates a model (or part of a model) from one BPMN or SysML format to an upgraded format.<br><br>Parameters: <ul style="list-style-type: none"> <li>GUID: String - the GUID of the package or element for which the contents are to be migrated</li> <li>SourceType: String - the type of model to be upgraded; accepted values:</li> </ul> |                                                                                                               |

| Method                                               | Type        | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also |
|------------------------------------------------------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                      |             | <ul style="list-style-type: none"> <li>• <b>BPMN</b></li> <li>• <b>BPMN1.1</b></li> <li>• <b>SysML1.1</b></li> </ul> <ul style="list-style-type: none"> <li>• DestinationType: String - the type of model to upgrade to; accepted values: <ul style="list-style-type: none"> <li>• <b>BPMN1.1</b></li> <li>• <b>BPMN1.1::BPEL</b></li> <li>• <b>BPMN2.0</b></li> <li>• <b>SysML1.2</b></li> </ul> </li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                              |          |
| <b>MigrateToBPMN11</b><br>(string GUID, string Type) | <i>Void</i> | <p>Migrates every BPMN 1.0 construct in a package or an element (including elements, attributes, diagrams and connectors) to BPMN 1.1.</p> <p>Parameters</p> <ul style="list-style-type: none"> <li>• GUID: String - the GUID of the package or element for which the contents are to be migrated to BPMN 1.1</li> <li>• Type: String - the type of upgrade, either just to BPMN 1.1 or to BPMN 1.1 and BPEL. Accepted values are: <p><b>BPMN</b> = migrate to BPMN 1.1<br/> <b>BPEL</b> = migrate to BPMN 1.1 and update:</p> <ul style="list-style-type: none"> <li>• any diagram with stereotype <i>BPMN</i> to <i>BPEL</i></li> <li>• any element with stereotype <i>BusinessProcess</i> to <i>BPELProcess</i></li> </ul> </li> </ul> <p>Migrating to BPEL is possible only in the Ultimate or Business and Software Engineering editions of Enterprise Architect.</p> |          |

| Method                                                                                      | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See also                                                                                                                                                                                     |
|---------------------------------------------------------------------------------------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ProjectTransfer</b> (string SourceFilePath, string TargetFilePath, string LogFilePath)   | <i>Boolean</i> | <p>Transfers the project from a .EAP file or DBMS to a .EAP file.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>SourceFilePath: String - the path of the source file to transfer</li> <li>TargetFilePath: String - the path of the target file; Enterprise Architect creates a new Base project in this location</li> <li>LogFilePath: String - the path of the log file where the status of the transfer process is updated</li> </ul> <p>In automation, the target file does not have to exist; the file path is enough. Enterprise Architect creates a new, empty Base.EAP file and transfers the source project into it.</p>                                                                                                                                                                                                                                                                |                                                                                                                                                                                              |
| <b>PublishResult</b> (string CategoryID, EA.EnumMVErrorType ErrorType, string ErrorMessage) | <i>String</i>  | <p>Returns the results of each rule that can be performed during model validation. It must be called once for each rule from the <i>EA_OnInitializeUserRules</i> broadcast handler.</p> <p>The return value is a <b>RuleId</b>, which can be used for reference purposes when an individual rule is executed by Enterprise Architect during model validation.</p> <p>See the <i>Model Validation Example</i> for a detailed example of the use of this method.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>CategoryId: String - should be passed the return value from the <i>DefineRuleCategory</i> method</li> <li>ErrorType: EA.EnumMVErrorType - depending on the severity of the error being validated, can be: <ul style="list-style-type: none"> <li><b>mvErrorCritical</b></li> <li><b>mvError</b></li> <li><b>mvWarning</b>, or</li> <li><b>mvInformation</b></li> </ul> </li> </ul> | <p><a href="#">EA_OnInitializeUserRules</a><sup>[2622]</sup></p> <p><a href="#">Model Validation Example</a><sup>[2630]</sup></p> <p><a href="#">DefineRuleCategory</a><sup>[2537]</sup></p> |



| Method                                                                         | Type                                  | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See also                                                                                                           |
|--------------------------------------------------------------------------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
|                                                                                |                                       | <ul style="list-style-type: none"> <li>ErrorMessage: String - contains an error string</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                    |
| <b>PutDiagramImageOnClipboard (string DiagramGUID, long Type)</b>              | protected abstract:<br><i>Boolean</i> | <p>Copies an image of the specified diagram to the clipboard.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to copy</li> <li>Type: Long - the file type <ul style="list-style-type: none"> <li>If Type = <b>0</b> then it is a metafile</li> <li>If Type = <b>1</b> then it is a Device Independent Bitmap</li> </ul> </li> </ul>                                                                                                                     |                                                                                                                    |
| <b>PutDiagramImageToFile (string Diagram GUID, string FileName, long Type)</b> | protected abstract:<br><i>Boolean</i> | <p>Saves an image of the specified diagram to file.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to save</li> <li>FileName: String - the name of the file to save the diagram into</li> <li>Type: Long - the file type <ul style="list-style-type: none"> <li>If type = <b>0</b> then it is a metafile</li> <li>If type = <b>1</b> then it uses the file type from the name extension (that is, .bmp, .jpg, .gif, .png, .tga)</li> </ul> </li> </ul> |                                                                                                                    |
| <b>ReloadProject ()</b>                                                        | protected abstract:<br><i>Boolean</i> | <p>Reloads the current project.</p> <p>This is a convenient method to refresh the current loaded project (in case of outside changes to the .EAP file).</p>                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                    |
| <b>RunModelSearch (string Search, string SearchTerm, bool ShowInEA)</b>        | <i>Void</i>                           | <p>Invokes the Model Search component.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Search: String - the name of an Enterprise Architect defined search</li> <li>SearchTerm: String - the term to search for in the project</li> </ul>                                                                                                                                                                                                                                                                             | <a href="#">Model Search</a>  |

| Method                                                                                                           | Type                               | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                       | See also |
|------------------------------------------------------------------------------------------------------------------|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                                                                                  |                                    | <ul style="list-style-type: none"> <li>ShowInEA: Boolean - execute the search and output in the Model Search window</li> </ul>                                                                                                                                                                                                                                                                                                                              |          |
| <b>RunReport (string PackageGUID, string TemplateName, string Filename)</b>                                      | protected abstract:<br><i>Void</i> | <p>Runs a named RTF report.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to run the report on</li> <li>TemplateName: String - the RTF report template to use; if the PackageGUID has a stereotype of <i>MasterDocument</i>, the template is not required</li> <li>FileName: String - the file name to store the generated report in</li> </ul>                               |          |
| <b>RunHTMLReport (string PackageGUID, string ExportPath, string ImageFormat, string Style, string Extension)</b> | <i>String</i>                      | <p>Runs an HTML report (as for <b>Documentation   HTML Documentation</b> when you right-click on a package in the Project Browser).</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to run the report on</li> <li>ExportPath: String - the file name to store the generated report in</li> <li>ImageFormat: String</li> <li>Style: String</li> <li>Extension: String</li> </ul> |          |
| <b>SaveControlledPackage (string PackageGUID)</b>                                                                | <i>String</i>                      | <p>Saves a package that has been configured as a controlled package, to XML. Only the package GUID is required, Enterprise Architect picks the rest up from the package control information.</p> <p>Parameter:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to save</li> </ul>                                                                                                                  |          |

| Method                                                              | Type                                 | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                  | See also |
|---------------------------------------------------------------------|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>SaveDiagramImageToFile (string Filename)</b>                     | protected abstract:<br><i>String</i> | Saves a diagram image of the current diagram to file.<br>Parameters: <ul style="list-style-type: none"> <li>• FileName: String - the filename of the image to save</li> </ul>                                                                                                                                                                                                                                                          |          |
| <b>ShowWindow (long Show)</b>                                       | protected abstract:<br><i>Void</i>   | Shows or hides the Enterprise Architect User Interface.<br>Parameters: <ul style="list-style-type: none"> <li>• Show: Long</li> </ul>                                                                                                                                                                                                                                                                                                  |          |
| <b>SynchronizeClass (string ElementGUID, string ExtraOptions)</b>   | <i>Boolean</i>                       | Synchronizes a Class with the latest source code.<br>Parameters: <ul style="list-style-type: none"> <li>• ElementGUID: String - the GUID (in XML format) of the element to update from code</li> <li>• ExtraOptions: String - enables extra options to be given to the command; currently unused</li> </ul>                                                                                                                            |          |
| <b>SynchronizePackage (string PackageGUID, string ExtraOptions)</b> | <i>Boolean</i>                       | Synchronizes each Class in a package with the latest source code.<br>Parameters: <ul style="list-style-type: none"> <li>• PackageGUID: String - the GUID (in XML format) of the package containing the elements to update from code</li> <li>• ExtraOptions: String - enables extra options to be given to the command; currently enables synchronization of all child packages (children) - for example: <i>children=1</i></li> </ul> |          |
| <b>TransformElement (string TransformName, string ElementGUID,</b>  | <i>Boolean</i>                       | Transforms an element into a package.<br>Parameters:                                                                                                                                                                                                                                                                                                                                                                                   |          |

| Method                                                                                                          | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See also |
|-----------------------------------------------------------------------------------------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>string TargetPackage, string ExtraOptions)</b>                                                               |                | <ul style="list-style-type: none"> <li>• TransformName: String - specifies the transformation that should be executed</li> <li>• ElementGUID: String - the GUID (in XML format) of the element to transform</li> <li>• TargetPackageGUID: String - the GUID (in XML format) of the package to transform into</li> <li>• ExtraOptions: String - enables extra options to be given to the command: <ul style="list-style-type: none"> <li>• GenCode=<b>True/False</b> - articulate code generation from the transformed elements; this option supercedes the current model setting</li> </ul> </li> </ul>                                                                              |          |
| <b>TransformPackage (string TransformName, string SourcePackage, string TargetPackage, string ExtraOptions)</b> | <i>Boolean</i> | <p>Runs a transformation on the contents of a package.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• TransformName: String - specifies the transformation that should be executed</li> <li>• SourcePackageGUID: String - the GUID (in XML format) of the package to transform</li> <li>• TargetPackageGUID: String - the GUID (in XML format) of the package to transform into</li> <li>• ExtraOptions: String - enables extra options to be given to the command: <ul style="list-style-type: none"> <li>• GenCode=true/false - articulate code generation from the transformed elements; this option supercedes the current model setting</li> </ul> </li> </ul> |          |

| Method                                      | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also                                  |
|---------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
|                                             |                | <ul style="list-style-type: none"> <li>SubPackages=true/false - specify if the child packages are to be included whilst transforming a package</li> </ul>                                                                                                                                                                                                                                                                                                                                                                  |                                           |
| <b>ValidateDiagram (string DiagramGUID)</b> | <i>Boolean</i> | <p>Invokes the Enterprise Architect Model Validation component, then validates the diagram (for correctness) and the elements and connectors within the diagram.</p> <p>Output can be viewed through <b>View   System Output   Model Validation</b>.</p> <p>Returns a boolean value to indicate the success or failure of the process, regardless of the results of the validation.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID of the Diagram Class object</li> </ul>    |                                           |
| <b>ValidateElement (string ElementGUID)</b> | <i>Boolean</i> | <p>Invokes the Enterprise Architect Model Validation component, then validates the element and all child elements, diagrams, connectors, attributes and operations.</p> <p>Output can be viewed through <b>View   System Output   Model Validation</b>.</p> <p>Returns a boolean value to indicate the success or failure of the process, regardless of the results of the validation.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID of the Element Class object</li> </ul> | <a href="#">Model Validation</a><br>22031 |
| <b>ValidatePackage (string PackageGUID)</b> | <i>Boolean</i> | <p>Invokes the Enterprise Architect Model Validation component, then validates the package and all sub-packages, elements, connectors and diagrams within it.</p> <p>Output can be viewed through <b>View   System Output   Model Validation</b>.</p> <p>Returns a boolean value to indicate the success or failure of the process, regardless of the results of the validation.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID of the Package Class object</li> </ul>       |                                           |

| Method                         | Type          | Notes                                                                                                                                                                                                                                       | See also |
|--------------------------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>XMLtoGUID (string GUID)</b> | <i>String</i> | <p>Changes a GUID in XML format to the form used inside Enterprise Architect.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>GUID: String - the XML style GUID to convert to Enterprise Architect internal format</li> </ul> |          |

**Notes**

- These methods all require input GUIDs in XML format; use *GUIDtoXML* to change the Enterprise Architect GUID to an XML GUID

**Learn more**

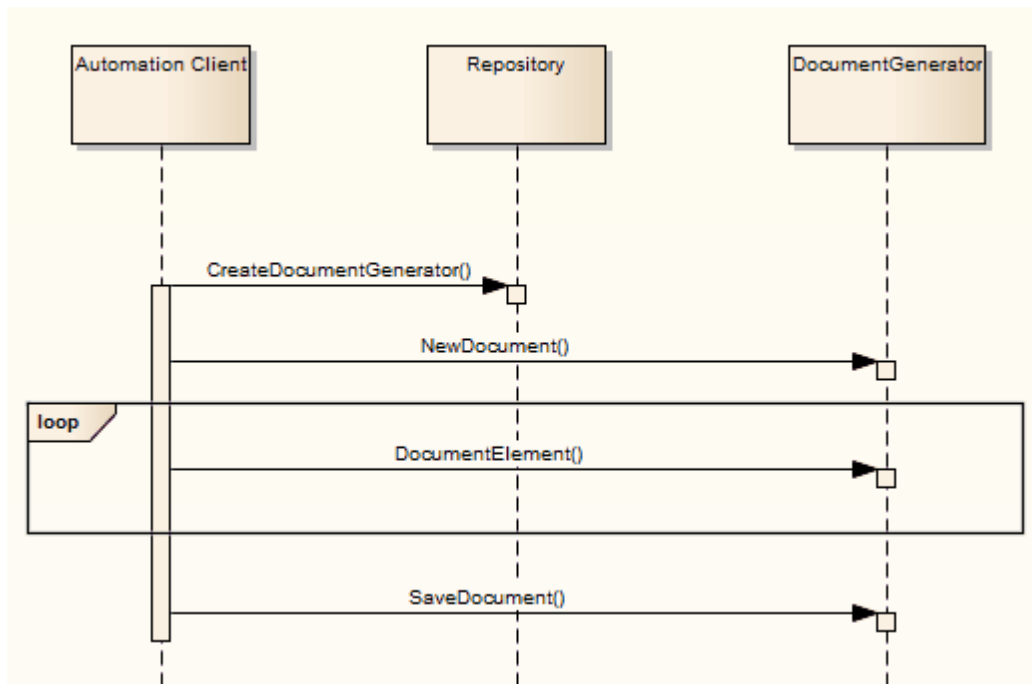
- [GUIDtoXML](#)<sup>[2549]</sup>

**19.2.2.9 Document Generator Interface Package**

The *Enterprise Architect.DocumentGenerator* interface. This is the XML-based interface to Enterprise Architect RTF and HTML reporting. You can get a pointer to this interface using the *Repository.CreateDocumentGenerator* method.

The following diagram illustrates how you might apply this interface in generating a report through the Automation Interface.

**Example**



#### 19.2.2.9.1 DocumentGenerator Class

The *DocumentGenerator* Class can be accessed from the Repository using *CreateProjectGenerator()*. The returned interface provides access to the XML-based Enterprise Architect Automation Interface. Use this interface to get XML for the various internal elements and to run some utility functions to perform tasks such as load diagrams or run reports.

##### DocumentGenerator Attributes:

| Attribute         | Type                                 | Notes                                                                       |
|-------------------|--------------------------------------|-----------------------------------------------------------------------------|
| <b>ObjectType</b> | <a href="#">ObjectType</a><br>[2419] | Read only<br>Distinguishes objects referenced through a Dispatch interface. |

##### Document Generator Methods:

| Method                                                                    | Type           | Notes                                                                                                                                                                                                                                                                                         |
|---------------------------------------------------------------------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DocumentDiagram (long diagramID, long nDepth, string templateName)</b> | <i>Boolean</i> | Documents a diagram.<br>Parameters: <ul style="list-style-type: none"> <li>• diagramId: Long - the ID of the diagram</li> <li>• nDepth: Long - adjust the heading level by nDepth</li> <li>• templateName: String - a template to use when documenting diagrams; this can be blank</li> </ul> |

| Method                                                                     | Type           | Notes                                                                                                                                                                                                                                                                                                         |
|----------------------------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DocumentElement (long elementID, long nDepth, string templateName)</b>  | <i>Boolean</i> | Documents an element.<br>Parameters: <ul style="list-style-type: none"> <li>• elementId: Long - the ID of the element</li> <li>• nDepth: Long - adjust the heading level by nDepth</li> <li>• templateName: String - a template to use when documenting elements; can be blank</li> </ul>                     |
| <b>DocumentModelAuthor (string name, long nDepth, string templateName)</b> | <i>Boolean</i> | Documents a model author.<br>Parameters: <ul style="list-style-type: none"> <li>• name: String - the name of the author</li> <li>• nDepth: Long - adjust the heading level by nDepth</li> <li>• templateName: String - a template to use when documenting model authors; can be blank</li> </ul>              |
| <b>DocumentModelClient (string name, long nDepth, string templateName)</b> | <i>Boolean</i> | Documents a single model client.<br>Parameters: <ul style="list-style-type: none"> <li>• name: String - the name of the client</li> <li>• nDepth: Long - adjust the heading level by nDepth</li> <li>• templateName: String - a template to use when documenting model clients; can be blank</li> </ul>       |
| <b>DocumentModelGlossary (long id, long nDepth, string templateName)</b>   | <i>Boolean</i> | Documents a single model glossary term.<br>Parameters: <ul style="list-style-type: none"> <li>• id: Long - the ID of the term</li> <li>• nDepth: Long - adjust the heading level by nDepth</li> <li>• templateName: String - a template to use when documenting model glossary terms; can be blank</li> </ul> |
| <b>DocumentModelIssue (long id, long nDepth, string templateName)</b>      | <i>Boolean</i> | Documents a single model issue.<br>Parameters: <ul style="list-style-type: none"> <li>• id: Long - the ID of the issue</li> <li>• nDepth: Long - adjust the heading level by nDepth</li> <li>• templateName: String - a template to use when documenting model issues; can be blank</li> </ul>                |
| <b>DocumentModelResource (string name, long nDepth,</b>                    | <i>Boolean</i> | Documents a single model resource.                                                                                                                                                                                                                                                                            |



| Method                                                                    | Type           | Notes                                                                                                                                                                                                                                                                                              |
|---------------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>string templateName)</b>                                               |                | Parameters: <ul style="list-style-type: none"> <li>• name: String - the name of the resource</li> <li>• nDepth: Long - adjust the heading level by nDepth</li> <li>• templateName: String - a template to use when documenting model resources; can be blank</li> </ul>                            |
| <b>DocumentModelRole (string name, long nDepth, string templateName)</b>  | <i>Boolean</i> | Documents a single model role.<br>Parameters: <ul style="list-style-type: none"> <li>• name: String - the name of the role</li> <li>• nDepth: Long - adjust the heading level by nDepth.</li> <li>• templateName: String - a template to use when documenting model roles; can be blank</li> </ul> |
| <b>DocumentModelTask (long id, long nDepth, string templateName)</b>      | <i>Boolean</i> | Documents a single model task.<br>Parameters: <ul style="list-style-type: none"> <li>• id: Long - the ID of the the task.</li> <li>• nDepth: Long - adjust the heading level by nDepth</li> <li>• templateName: String - a template to use when documenting model tasks; can be blank</li> </ul>   |
| <b>DocumentPackage (long packageID, long nDepth, string templateName)</b> | <i>Boolean</i> | Documents a package.<br>Parameters: <ul style="list-style-type: none"> <li>• packageId: Long - the ID of the package</li> <li>• nDepth: Long - adjust the heading level by nDepth</li> <li>• templateName: String - a template to use when documenting packages; can be blank</li> </ul>           |
| <b>GetLastError()</b>                                                     | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.                                                                                                                                                                                                  |
| <b>InsertBreak (long breakType)</b>                                       | <i>Boolean</i> | Inserts a break into the report at the current location.<br>Parameters: <ul style="list-style-type: none"> <li>• breakType: <b>0</b> = page break, <b>1</b> = section break</li> </ul>                                                                                                             |
| <b>InsertLinkedDocument (string guid)</b>                                 | <i>Boolean</i> | Inserts a linked document into the report at the current location.<br><br>A linked document can used to set the header and footer of                                                                                                                                                               |

| Method                                                    | Type           | Notes                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                           |                | <p>the report. These are taken from the first linked document added to the report.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>guid: String - the GUID of the element that has a linked document</li> </ul>                                                                                                                                       |
| <b>InsertTableOfContents</b>                              | <i>Boolean</i> | Inserts a Table of Contents at the current position.                                                                                                                                                                                                                                                                                                                |
| <b>InsertTeamReviewPost (string path)</b>                 | <i>Boolean</i> | <p>Inserts a Team Review posting into the report at the current location.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>path: String - the path of the Team Review post</li> </ul>                                                                                                                                                                  |
| <b>InsertText (string text, string style)</b>             | <i>Boolean</i> | <p>Inserts some static text into the report at the current location.</p> <p>No carriage return is added, so remember to add one if required.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>text: String - the static text to be inserted</li> <li>style: String - the name of the style in the template; defaults to <i>Normal</i> style</li> </ul> |
| <b>NewDocument (string templateName)</b>                  | <i>Boolean</i> | <p>Starts a new document; this must be called before attempting to document anything else.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>templateName: String - a template to use when documenting elements; this can be blank</li> </ul>                                                                                                           |
| <b>ReplaceField (string fieldname, string fieldvalue)</b> | <i>Boolean</i> | <p>Updates the fieldname with the value given.</p> <p>If you call this function more than once with the same fieldname, the field only has the last value set.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>fieldname: String - the field name to find</li> <li>fieldvalue: String - the value to insert into the field</li> </ul>                 |
| <b>SaveDocument (string filename, long nDocType)</b>      | <i>Boolean</i> | <p>Saves the document to disk.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>filename: String - the filename to save the file to</li> <li>nDocType: <b>0</b> = RTF, <b>1</b> = HTML</li> </ul>                                                                                                                                                      |

| Method                                           | Type           | Notes                                                                                                                                            |
|--------------------------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SetPageOrientation (long pageOrientation)</b> | <i>Boolean</i> | Sets the current page orientation.<br>Parameters: <ul style="list-style-type: none"> <li>pageOrientation: 0 = Portrait, 1 = Landscape</li> </ul> |

### 19.2.2.10 Mail Interface Package

The Enterprise Architect *MailInterface* package contains:

- A function to retrieve a pointer to the interface
- Functions to create and send a mail message within the current mode
- Utility functions for creating hyperlinks to selected model elements

You can get a pointer to this interface using the method *Repository.GetMailInterface*.

#### Learn more

- [Repository Package](#) [2423]
- [MailInterface Class](#) [2566]

#### 19.2.2.10.1 MailInterface Class

The *MailInterface* interface can be accessed from the Repository using *GetMailInterface()*. The returned interface provides access to the Enterprise Architect Model Mail Interface. Use this interface to automate the process of creating and sending messages using Enterprise Architect's Model Mail system.

#### MailInterface Attributes:

| Attribute               | Type                                                             | Notes                                                                       |
|-------------------------|------------------------------------------------------------------|-----------------------------------------------------------------------------|
| <b>ObjectType</b>       | <a href="#">ObjectTyp</a> <small>e</small> <small>[2419]</small> | Read Only<br>Distinguishes objects referenced through a dispatch interface. |
| <b>MessagingEnabled</b> | Boolean                                                          | Read Only<br>Advises whether messaging is enabled on the current model.     |

#### MailInterface Methods:

| Method                  | Type          | Notes                                                     |
|-------------------------|---------------|-----------------------------------------------------------|
| <b>GetLastError ( )</b> | <i>String</i> | Returns the last error message set for the MailInterface. |

| Method                                                                  | Type          | Notes                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------------------------------------------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetRecipientGUID</b><br>(string UserName)                            | <i>String</i> | Returns the GUID of the specified Enterprise Architect user.<br>Parameters: <ul style="list-style-type: none"> <li>• UserName: String - The name of a user defined in the current model</li> </ul>                                                                                                                                                            |
| <b>GetElementHyperlink</b><br>(string ElementGUID, string LinkText)     | <i>String</i> | Returns a string containing a hyperlink to the element specified by the input parameter ElementGUID.<br>Parameters: <ul style="list-style-type: none"> <li>• ElementGUID: String - The GUID of the element for which a hyperlink is required</li> <li>• LinkText: String - The text to display for the hyperlink (e.g. the element name)</li> </ul>           |
| <b>GetAttributeHyperlink</b><br>(string AttributeGUID, string LinkText) | <i>String</i> | Returns a string containing a hyperlink to the attribute specified by the input parameter AttributeGUID.<br>Parameters: <ul style="list-style-type: none"> <li>• AttributeGUID: String - The GUID of the attribute for which a hyperlink is required</li> <li>• LinkText: String - The text to display for the hyperlink (e.g. the attribute name)</li> </ul> |
| <b>GetMethodHyperlink</b><br>(string MethodGUID, string LinkText)       | <i>String</i> | Returns a string containing a hyperlink to the method specified by the input parameter MethodGUID.<br>Parameters: <ul style="list-style-type: none"> <li>• MethodGUID: String - The GUID of the method for which a hyperlink is required</li> <li>• LinkText: String - The text to display for the hyperlink (e.g. the method name)</li> </ul>                |
| <b>GetDiagramHyperlink</b><br>(string DiagramGUID, string LinkText)     | <i>String</i> | Returns a string containing a hyperlink to the diagram specified by the input parameter DiagramGUID.<br>Parameters: <ul style="list-style-type: none"> <li>• DiagramGUID: String - The GUID of the diagram for which a hyperlink is required</li> <li>• LinkText: String - The text to display for the hyperlink (e.g. the diagram name)</li> </ul>           |

| Method                                                                                                                            | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------------------------------------------------------------------------------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetPackageHyperlink(string PackageGUID, string LinkText)</b>                                                                   | <i>String</i>  | Returns a string containing a hyperlink to the package specified by the input parameter PackageGUID.<br>Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - The GUID of the package for which a hyperlink is required</li> <li>LinkText: String - The text to display for the hyperlink (e.g. the package name)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                    |
| <b>GetFileHyperlink(string FilePath, string LinkText)</b>                                                                         | <i>String</i>  | Returns a string containing a hyperlink to the file specified by the input parameter FilePath.<br>Parameters: <ul style="list-style-type: none"> <li>FilePath: String - The path name of the file for which a hyperlink is required</li> <li>LinkText: String - The text to display for the hyperlink (e.g. the file name)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                              |
| <b>GetWebHyperlink(string URL, string LinkText)</b>                                                                               | <i>String</i>  | Returns a string containing a hyperlink to the URL specified by the input parameter URL.<br>Parameters: <ul style="list-style-type: none"> <li>URL: String - The URL of the item for which a hyperlink is required</li> <li>LinkText: String - The text to display for the hyperlink</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>ComposeMailMessage(string InitialRecipientGUID, string InitialSubject, messageflag InitialFlag, string InitialMessageText)</b> | <i>Boolean</i> | Creates a new mail message using the values specified in the input parameters; the message is displayed in the composition window, ready for sending.<br>This method does NOT send the message.<br>Parameters: <ul style="list-style-type: none"> <li>InitialRecipientGUID: String - Initial value for the GUID of the addressee user (an Enterprise Architect user defined in the current model)</li> <li>InitialSubject: String - Initial value for the Subject text to display for this message</li> <li>InitialFlag: MessageFlag - Initial value for the flag type/color to attach to this message</li> <li>InitialMessageText: String - Initial value for the text that is the body of the message</li> </ul> |
| <b>SendMailMessage</b>                                                                                                            | <i>Boolean</i> | Creates and sends a new mail message using the values specified in                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

| Method                                                                              | Type | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>(string RecipientGUID, string Subject, messageflag Flag, string MessageText)</b> |      | <p>the input parameters.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>RecipientGUID: String - The GUID of the addressee user (an Enterprise Architect user defined in the current model)</li> <li>Subject: String - The Subject text to display for this message</li> <li>Flag: MessageFlag - The flag type/color to attach to this message</li> <li>MessageText: String - The text that is the body of the message</li> </ul> |

### 19.2.2.11 Code Samples

This topic contains various code examples indicating how to use the Automation Interface, written in VB.Net:

| Topic                       | Link                                                                                                                                     |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Open the Repository         | <a href="#">Open the Repository</a>  [2570]           |
| Iterate Through a .EAP File | <a href="#">Iterate Through a .EAP File</a>  [2570] |
| Add and Manage Packages     | <a href="#">Add and Manage Packages</a>  [2571]     |
| Add and Manage Elements     | <a href="#">Add and Manage Elements</a>  [2572]     |
| Add a Connector             | <a href="#">Add a Connector</a>  [2573]              |
| Add and Manage Diagrams     | <a href="#">Add and Manage Diagrams</a>  [2574]     |
| Add and Delete Features     | <a href="#">Add and Delete Features</a>  [2575]     |
| Element Extras              | <a href="#">Element Extras</a>  [2575]               |
| Repository Extras           | <a href="#">Repository Extras</a>  [2578]           |
| Stereotypes                 | <a href="#">Stereotypes</a>  [2580]                   |
| Work with Attributes        | <a href="#">Work with Attributes</a>  [2581]        |

| Topic             | Link                                                |
|-------------------|-----------------------------------------------------|
| Work with Methods | <a href="#">Work with Methods</a> <sup>[2582]</sup> |

### 19.2.2.11.1 Open the Repository

#### Topics

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> '' An example of how to open an Enterprise Architect repository '' in VB. Net .  Public Class AutomationExample      ' class level variable for Repository     Public m_Repository As Object      Public Sub Run()         try             ' create the repository object             m_Repository = CreateObject("EA.Repository")              ' open an EAP file             m_Repository.OpenFile("F:\Test\EAAuto.EAP")             ' use the Repository in any way required             ' DumpModel              ' close the repository and tidy up             m_Repository.Exit()             m_Repository = Nothing              ... catch e as exception                 Console.WriteLine(e)             End try         End Sub     end Class </pre> |

### 19.2.2.11.2 Iterate Through a .EAP File

#### Topics

| Topic                | Detail                                                                                                                                                                                                                                          |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> '' Assume repository has already been opened.  '' Start at the model level Sub DumpModel()     Dim idx as Integer     For idx=0 to m_Repository.Models.Count - 1         DumpPackage("", m_Repository.Models.GetAt (idx))     Next </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> End Sub  ' output package name, then element contents, then process child packages Sub DumpPackage(Indent as String, Package as Object)     Dim idx as Integer     Console.WriteLine(Indent + Package.Name)     DumpElements(Indent + "    ", Package)      For idx = 0 to Package.Packages.Count - 1         DumpPackage(Indent + "    ", Package. Packages.GetAt(idx))     Next End Sub  '' dump element name Sub DumpElements(Indent as String, Package as Object)     Dim idx as Integer     For idx = 0 to Package.Elements.Count - 1         Console.WriteLine(Indent + "::" + Package. Elements.GetAt(idx).Name)     Next End Sub </pre> |

### 19.2.2.11.3 Add and Manage Packages

#### Topics

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> Example illustrating how to add a Model or a Package.  Sub TestPackageLifecycle      Dim idx as integer     Dim idx2 as integer     Dim package as object     Dim model as object     Dim o as object      '' first add a new Model     model = m_Repository.Models.AddNew ("AdvancedModel", "")     If not model.Update() Then         Console.WriteLine(model.GetLastError())     End If      '' refresh the models collection     m_Repository.Models.Refresh      '' now work through models collection and add a package      For idx = 0 to m_Repository.Models.Count - 1         o = m_Repository.Models.GetAt(idx)         Console.WriteLine(o.Name)         If o.Name = "AdvancedModel" Then             package = o.Packages.AddNew ("Subpackage", "Nothing") </pre> |



| Topic | Detail                                                                                                                                                                                                                                                                                       |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> If not package.Update() Then     Console.WriteLine(package. Get Last Error()) End If  package.Element.Stereotype = "system" package.Update  ''for testing purposes just delete the ''newly created Model and its contents m_Repository.Models.Delete(idx)  End If Next  End Sub </pre> |

#### 19.2.2.11.4 Add and Manage Elements

##### Topics

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> '' Add and delete elements in a package.  Sub ElementLifecycle      Dim package as Object     Dim element as Object      package = m_Repository.GetPackageByID(2)     element = package.elements.AddNew("Login to Website", "UseCase")     element.Stereotype = "testcase"     element.Update     package.elements.Refresh()      Dim idx as integer      '' note the repeated calls to "package.elements. Get At "     '' in general you should make this call once and assign to a local     '' variable - in the example below, Enterprise Architect loads the     '' element required every time a call is made - rather than loading once     '' and keeping a local reference.      For idx = 0 to package.elements.count - 1         Console.WriteLine(package.elements.Get At (idx). Name)         If (package.elements.Get At (idx).Name = "Login to Website" and _             package.elements.Get At (idx).Type = "UseCase") Then             package.elements.deleteat (idx, false)         End If     End If </pre> |

| Topic | Detail                    |
|-------|---------------------------|
|       | <pre> Next End Sub </pre> |

### 19.2.2.11.5 Add a Connector

#### Topics

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> " Add a connector and set values.  Sub ConnectorTest      Dim source as object     Dim target as object     Dim con as object     Dim o as object      Dim client as object     Dim supplier as object      '' use ElementIDs to quickly load an element in this example     '' ... you must find suitable IDs in your model      source = m_Repository.GetElementByID(129)     target = m_Repository.GetElementByID(169)      con = source.Connectors.AddNew ("test link 2", " Association")      '' again- replace ID with a suitable one from your model     con.SupplierID = 169      If not con.Update Then         Console.WriteLine(con.GetLastError)     End If     source.Connectors.Refresh      Console.WriteLine(" Connector Created")      o = con.Constraints.AddNew ("constraint2", "type")     If not o.Update Then         Console.WriteLine(o.GetLastError)     End If      o = con.TaggedValues.AddNew ("Tag", "Value")     If not o.Update Then         Console.WriteLine(o.GetLastError)     End If      '' use the client and supplier ends to set '' additional information      client = con.ClientEnd     client.Visibility = "Private"     client.Role = "m_client"     client.Update     supplier = con.SupplierEnd     supplier.Visibility = "Protected" </pre> |

| Topic | Detail                                                                                                                                                                            |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> supplier.Role = "m_supplier" supplier.Update  Console.WriteLine("Client and Supplier set")  Console.WriteLine(client.Role) Console.WriteLine(supplier.Role)  End Sub </pre> |

### 19.2.2.11.6 Add and Manage Diagrams

#### Topics

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> '' An example of how to create a diagram and add an element to it. '' Note the optional use of element rectangle setting using '' left, right, top and bottom dimensions in AddNew call.  Sub DiagramLifeCycle      Dim diagram as object     Dim v as object     Dim o as object     Dim package as object      Dim idx as Integer     Dim idx2 as integer      package = m_Repository.GetPackageById(5)      diagram = package.Diagrams.AddNew("Logical Diagram", "Logical")     If not diagram.Update Then         Console.WriteLine(diagram.LastError)     End If      diagram.Notes = "Hello there this is a test"     diagram.update()      o = package.Elements.AddNew ("ReferenceType", "Class")     o.Update      '' add element to diagram - supply optional rectangle co-ordinates      v = diagram.DiagramObjects.AddNew("l=200;r=400; t=200;b=600;", "")     v.ElementID = o.ElementID     v.Update      Console.WriteLine(diagram.DiagramID)  End Sub </pre> |

### 19.2.2.11.7 Add and Delete Features

#### Topics

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> Dim element as object Dim idx as integer Dim attribute as object Dim method as object  'just load an element by ID - you must 'substitute a valid ID from your model element = m_Repository.GetElementById(246)  '' create a new method method = element.Methods.AddNew("newMethod", "int") method.Update element.Methods.Refresh  'now loop through methods for Element - and delete our addition For idx = 0 to element.Methods.Count - 1     method = element.Methods.GetAt(idx)     Console.WriteLine(method.Name)     If (method.Name = "newMethod") Then         element.Methods.Delete(idx)     End If Next  ' create an attribute attribute = element.attributes.AddNew("NewAttribute", "int") attribute.Update element.attributes.Refresh  'loop through and delete our new attribute For idx = 0 to element.attributes.Count - 1     attribute = element.attributes.GetAt(idx)     Console.WriteLine(attribute.Name)     If (attribute.Name = "NewAttribute") Then         element.attributes.Delete(idx)     End If Next </pre> |

### 19.2.2.11.8 Element Extras

#### Topics

| Topic                | Detail                                                                                                                                                                                                         |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> '' Examples of how to access and use element extras, such as '' scenarios, constraints and requirements.  Sub ElementExtras      Dim element as object     Dim o as object     Dim idx as Integer </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> Dim bDel as boolean bDel = true  try     element = m_Repository.GetElementByID(129)      'manage constraints for an element     'demonstrate addnew and delete     o = element.Constraints.AddNew ("Appended", "Type")     If not o.Update Then         Console.WriteLine("Constraint error:" + o. GetLastError())     End if     element.Constraints.Refresh     For idx = 0 to element.Constraints.Count - 1         o = element.Constraints.GetAt(idx)         Console.WriteLine(o.Name)         If(o.Name="Appended") Then             If bDel Then element.Constraints.Delete (idx)         End if     Next      ' efforts     o = element.Efforts.AddNew("Appended", "Type")     If not o.Update Then         Console.WriteLine("Efforts error:" + o. GetLastError())     End if     element.Efforts.Refresh     For idx = 0 to element.Efforts.Count - 1         o = element.Efforts.GetAt(idx)         Console.WriteLine(o.Name)         If(o.Name="Appended") Then             If bDel Then element.Efforts.Delete (idx)         End if     Next      ' Risks     o = element.Risks.AddNew("Appended", "Type")     If not o.Update Then         Console.WriteLine("Risks error:" + o. GetLastError())     End if     element.Risks.Refresh     For idx = 0 to element.Risks.Count - 1         o = element.Risks.GetAt(idx)         Console.WriteLine(o.Name)         If(o.Name="Appended") Then             If bDel Then element.Risks.Delete(idx)         End if     Next      ' Metrics     o = element.Metrics.AddNew("Appended", "Change")     If not o.Update Then         Console.WriteLine("Metrics error:" + o. GetLastError())     End if     element.Metrics.Refresh     For idx = 0 to element.Metrics.Count - 1         o = element.Metrics.GetAt(idx)         Console.WriteLine(o.Name) </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre>                 If (o.Name="Appended") Then                     If bDel Then element.Metrics.Delete                 (idx)                 End if             Next              ' TaggedValues             o = element.TaggedValues.AddNew             ("Appended", "Change")             If not o.Update Then                 Console.WriteLine("TaggedValues error:" + o.GetLastError())             End if             element.TaggedValues.Refresh             For idx = 0 to element.TaggedValues.Count - 1                 o = element.TaggedValues.GetAt(idx)                 Console.WriteLine(o.Name)                 If(o.Name="Appended") Then                     If bDel Then element.TaggedValues. Delete (idx)                 End if             Next              ' Scenarios             o = element.Scenarios.AddNew             ("Appended", "Change")             If not o.Update Then                 Console.WriteLine("Scenarios error:" + o. GetLastError())             End if             element.Scenarios.Refresh             For idx = 0 to element.Scenarios.Count - 1                 o = element.Scenarios.GetAt(idx)                 Console.WriteLine(o.Name)                 If(o.Name="Appended") Then                     If bDel Then element.Scenarios.Delete                 (idx)                 End if             Next              ' Files             o = element.Files.AddNew("MyFile", "doc")             If not o.Update Then                 Console.WriteLine("Files error:" + o. GetLastError())             End if             element.Files.Refresh             For idx = 0 to element.Files.Count - 1                 o = element.Files.GetAt(idx)                 Console.WriteLine(o.Name)                 If(o.Name="MyFile") Then                     If bDel Then element.Files.Delete (idx)                 End if             Next              ' Tests             o = element.Tests.AddNew("TestPlan", "Load")             If not o.Update Then                 Console.WriteLine("Tests error:" + o. GetLastError())             End if             element.Tests.Refresh             For idx = 0 to element.Tests.Count - 1                 o = element.Tests.GetAt(idx)                 Console.WriteLine(o.Name) </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre>                 If (o.Name="Test Plan") Then                     If bDel Then element.Tests.Delete (idx)                 End if             Next              ' Defect             o = element.Issues.AddNew(" Broken", " Defect ")             If not o.Update Then                 Console.WriteLine("Issues error:" + o. Get Last Error())             End if             element.Issues.Refresh             For idx = 0 to element.Issues.Count - 1                 o = element.Issues.Get At (idx)                 Console.WriteLine(o.Name)                 If(o.Name="Broken") Then                     If bDel Then element.Issues.Delete (idx)                 End if             Next              ' Change             o = element.Issues.AddNew(" Change", " Change ")             If not o.Update Then                 Console.WriteLine("Issues error:" + o. Get Last Error())             End if             element.Issues.Refresh             For idx = 0 to element.Issues.Count - 1                 o = element.Issues.Get At (idx)                 Console.WriteLine(o.Name)                 If(o.Name="Change") Then                     If bDel Then element.Issues.Delete (idx)                 End if             Next              catch e as exception                 Console.WriteLine(element.Methods.Get Last Error ())                 Console.WriteLine(e)             End try          End Sub </pre> |

### 19.2.2.11.9 Repository Extras

#### Topics

| Topic                | Detail                                                                                                                                                                            |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> '' Examples of how to access repository '' collections for system level information.  Sub RepositoryExtras      Dim o as object     Dim idx as integer      ' issues </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> o = m_Repository.Issues.AddNew("Problem", "Type") If(o.Update=false) Then     Console.WriteLine(o.GetLastError()) End if o = nothing m_Repository.Issues.Refresh For idx = 0 to m_Repository.Issues.Count - 1     Console.WriteLine(m_Repository.Issues.GetAt (idx).Name)     If(m_Repository.Issues.GetAt(idx).Name = "Problem") then         m_Repository.Issues.DeleteAt(idx, false)         Console.WriteLine("Delete Issues")     End if Next  'tasks o = m_Repository.Tasks.AddNew("Task 1", "Task type") If(o.Update=false) Then     Console.WriteLine("error - " + o.GetLastError ()) End if o = nothing m_Repository.Tasks.Refresh For idx = 0 to m_Repository.Tasks.Count - 1     Console.WriteLine(m_Repository.Tasks.GetAt (idx).Name)     If(m_Repository.Tasks.GetAt(idx).Name = "Task 1") then         m_Repository.Tasks.DeleteAt(idx, false)         Console.WriteLine("Delete Tasks")     End if Next  'glossary o = m_Repository.Terms.AddNew("Term 1", "business") If(o.Update=false) Then     Console.WriteLine("error - " + o.GetLastError ()) End if o = nothing m_Repository.Terms.Refresh For idx = 0 to m_Repository.Terms.Count - 1     Console.WriteLine(m_Repository.Terms.GetAt (idx).Term)     If(m_Repository.Terms.GetAt(idx).Term = "Term 1") then         m_Repository.Terms.DeleteAt(idx, false)         Console.WriteLine("Delete Terms")     End if Next  'authors o = m_Repository.Authors.AddNew("Joe B", "Writer") If(o.Update=false) Then     Console.WriteLine(o.GetLastError()) End if o = nothing m_Repository.Authors.Refresh For idx = 0 to m_Repository.authors.Count - 1     Console.WriteLine(m_Repository.Authors.GetAt (idx).Name)     If(m_Repository.authors.GetAt(idx).Name = "Joe B") then         m_Repository.authors.DeleteAt(idx, false)         Console.WriteLine("Delete Authors")     End if </pre> |



| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> Next      o = m_Repository.Clients.AddNew("Joe Sphere", "Client")     If (o.Update=false) Then         Console.WriteLine(o.GetLastError())     End if     o = nothing     m_Repository.Clients.Refresh     For idx = 0 to m_Repository.Clients.Count - 1         Console.WriteLine(m_Repository.Clients.GetAt (idx).Name)         If(m_Repository.Clients.GetAt(idx).Name = "Joe Sphere") then             m_Repository.Clients.DeleteAt(idx,false)             Console.WriteLine("Delete Clients")         End if     Next      o = m_Repository.Resources.AddNew("Joe Worker", "Resource")     If (o.Update=false) Then         Console.WriteLine(o.GetLastError())     End if     o = nothing     m_Repository.Resources.Refresh     For idx = 0 to m_Repository.Resources.Count - 1         Console.WriteLine(m_Repository.Resources.GetAt (idx).Name)         If(m_Repository.Resources.GetAt(idx).Name = "Joe Worker") then             m_Repository.Resources.DeleteAt(idx,false)             Console.WriteLine("Delete Resources")         End if     Next  End Sub </pre> |

### 19.2.2.11.10 Stereotypes

#### Topics

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> Sub Test Stereotypes      Dim o as object     Dim idx as integer      '' add a new stereotype to the Stereotypes collection     o = m_Repository.Stereotypes.AddNew ("funky", "class")     If (o.Update=false) Then         Console.WriteLine(o.GetLastError())     End if     o = nothing      '' make sure you refresh </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> m_Repository.Stereotypes.Refresh  '' then iterate through - deleting our new entry in the process For idx = 0 to m_Repository.Stereotypes.Count - 1   Console.WriteLine(m_Repository.Stereotypes. GetAt(idx).Name)   If (m_Repository.Stereotypes.GetAt(idx).Name = "funky") then     m_Repository.Stereotypes.DeleteAt(idx, false)   Console.WriteLine("Delete element") End if Next End Sub </pre> |

### 19.2.2.11.11 Work With Attributes

#### Topics

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> '' An example of working with attributes.  Sub AttributeLifecycle    Dim element as object   Dim o as object   Dim t as object   Dim idx as Integer   Dim idx2 as integer   try     element = m_Repository.GetElementById(129)      For idx = 0 to element.Attributes.Count - 1        Console.WriteLine("attribute=" + element. Attributes.GetAt(idx).Name)        o = element.Attributes.GetAt(idx)       t = o.Constraints.AddNew("&gt; 123", "Precision")       t.Update()       o.Constraints.Refresh       For idx2 = 0 to o.Constraints.Count - 1         t = o.Constraints.GetAt(idx2)         Console.WriteLine("Constraint: " + t. Name)         If (t.Name="&gt; 123") Then           o.Constraints.DeleteAt(idx2, false)         End if       Next        For idx2 = 0 to o.TaggedValues.Count - 1         t = o.TaggedValues.GetAt(idx2)         If (t.Name = "Type2") Then           Console.WriteLine("deleting")           o.TaggedValues.DeleteAt(idx2, true)         End if       Next     end try   end sub </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> t = o.TaggedValues.AddNew("Type2", "Number") t.Update o.TaggedValues.Refresh For idx2 = 0 to o.TaggedValues.Count - 1     t = o.TaggedValues.GetAt(idx2)     Console.WriteLine("Tagged Value: " + t. Name) Next  If (element.Attributes.GetAt(idx).Name = "m_Tootle") Then     Console.WriteLine("delete attribute")     element.Attributes.DeleteAt(idx, false) End If  Next  catch e as exception     Console.WriteLine(element.Attributes. GetLastError())     Console.WriteLine(e) End try End Sub </pre> |

### 19.2.2.11.12 Work With Methods

#### Topics

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> '' An example of working with the Methods collection '' of an element - and with Method collections.  Sub MethodLifeCycle      Dim element as object     Dim method as object     Dim t as object     Dim idx as Integer     Dim idx2 as integer      try         element = m_Repository.GetElementByID(129)          For idx = 0 to element.Methods.Count - 1             method = element.Methods.GetAt(idx)             Console.WriteLine(method.Name)              t = method.PreConditions.AddNew ("Test Constraint", "something")             If t.Update = false Then                 Console.WriteLine("PreConditions: " + t.GetLastError)             End if              method.PreConditions.Refresh             For idx2 = 0 to method.PreConditions.Count - 1 </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> t = method.PreConditions.GetAt(idx2) Console.WriteLine("PreConditions: " + t.Name) If t.Name = "TestConstraint" Then method.PreConditions.DeleteAt(idx2, false) End If Next  t = method.PostConditions.AddNew ("TestConstraint", "something") If t.Update = false Then Console.WriteLine("PostConditions: " + t.GetLastError) End If  method.PostConditions.Refresh For idx2 = 0 to method.PostConditions. Count - 1 t = method.PostConditions.GetAt(idx2) Console.WriteLine("PostConditions: " + t.Name) If t.Name = "TestConstraint" Then method.PostConditions.DeleteAt (idx2, false) End If Next  t = method.TaggedValues.AddNew ("TestTaggedValue", "something") If t.Update = false Then Console.WriteLine("Tagged Values: " + t.GetLastError) End If  For idx2 = 0 to method.TaggedValues.Count - 1 t = method.TaggedValues.GetAt(idx2) Console.WriteLine("Tagged Value: " + t. Name) If(t.Name= "TestTaggedValue") Then method.TaggedValues.DeleteAt(idx2, false) End If Next  t = method.Parameters.AddNew ("TestParam", "string") If t.Update = false Then Console.WriteLine("Parameters: " + t. GetLastError) End If  method.Parameters.Refresh For idx2 = 0 to method.Parameters.Count - 1 t = method.Parameters.GetAt(idx2) Console.WriteLine("Parameter: " + t. Name) If(t.Name="TestParam") Then method.Parameters.DeleteAt(idx2, false) End If Next  method = nothing Next catche as exception Console.WriteLine(element.Methods.GetLastError </pre> |

| Topic | Detail                                                                                                           |
|-------|------------------------------------------------------------------------------------------------------------------|
|       | <pre data-bbox="528 309 975 439">    ( )<br/>        Console.WriteLine(e)<br/>    End try<br/><br/>End Sub</pre> |

## 19.3 Enterprise Architect Add-In Model








### Topics

| Topic               | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See also                                                                                                                                                                                                                                                                                |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Introduction</b> | <p>Add-Ins enable you to add functionality to Enterprise Architect. The Enterprise Architect Add-In model builds on the features provided by the <b>Automation Interface</b> to enable you to extend the Enterprise Architect user interface.</p> <p>Add-Ins are ActiveX COM objects that expose public Dispatch methods. They have several advantages over stand-alone automation clients:</p> <ul style="list-style-type: none"> <li>• Add-Ins can define Enterprise Architect menus and sub-menus</li> <li>• Add-Ins receive notifications about various Enterprise Architect user-interface events including menu clicks and file changes</li> <li>• Add-Ins can (and should) be written as in-process (DLL) components. This provides lower call overhead and better integration into the Enterprise Architect environment</li> <li>• Because a current version of Enterprise Architect is already running there is no requirement to start a second copy of Enterprise Architect via the automation interface</li> <li>• Because the Add-In receives object handles associated with the currently running copy of Enterprise Architect, more information is available about the current user's activity; for example, which diagram objects are selected</li> <li>• You are not required to do anything other than to install the Add-In to make it usable; that is, you do not have to configure Add-Ins to run on your systems</li> <li>• Because Enterprise Architect is constantly evolving in response to customer requests, the Add-In interface is flexible:</li> <li>• The Add-In interface does not have its own version, rather it is identified by the version of Enterprise Architect it first appeared in; for example, the current version of the Enterprise Architect Add-In interface is version 2.1</li> <li>• When creating your Add-In, you do not have to subscribe to a type-library</li> </ul> | <p><a href="#">Automation Interface</a> <sup>[2381]</sup></p> <p><a href="#">Scripts</a> <sup>[2372]</sup></p> <p><a href="#">UML Profiles</a> <sup>[1293]</sup></p> <p><a href="#">Registered Add-in</a> <sup>[2745]</sup></p> <p><a href="#">Add-In Manager</a> <sup>[2593]</sup></p> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>From Enterprise Architect release 7.0, Add-Ins created before 2004 are no longer supported. If an Add-In subscribes to the Addn_Tmpl.tlb interface (2003 style), it fails on load. In this event, contact the vendor or author of the Add-In and request an upgrade.</p> <ul style="list-style-type: none"> <li>• Add-Ins do not have to implement methods that they never use</li> <li>• Add-Ins prompt users via context menus in the tree view and the diagram</li> <li>• Menu check and disable states can be controlled by the Add-In</li> </ul> <p>Add-Ins enhance the existing functionality of Enterprise Architect through a variety of mechanisms such as <b>Scripts</b>, <b>UML Profiles</b> and the <b>Automation Interface</b>. Once an <b>Add-In</b> is registered, it can be managed using the <b>Add-In Manager</b>.</p> |          |




### Create and Use Add-Ins:

This topic covers the following information on Add-Ins:

| Topic            | Link                                                                                                                   |
|------------------|------------------------------------------------------------------------------------------------------------------------|
| Add-In Tasks     | <a href="#">Add-In Tasks</a>      |
| Add-In Events    | <a href="#">Add-In Events</a>     |
| Broadcast Events | <a href="#">Broadcast Events</a>  |
| Custom Views     | <a href="#">Custom Views</a>      |
| MDG Add-Ins      | <a href="#">MDG Add-Ins</a>       |

### **19.3.1 Add-In Tasks**

This topic provides instructions on how to create, test, deploy and manage Add-Ins.

| Topic                   | Detail                  | Link                                                                                                                         |
|-------------------------|-------------------------|------------------------------------------------------------------------------------------------------------------------------|
| <b>Create an Add-In</b> |                         | <a href="#">Create an Add-In</a>        |
|                         | Define Menu Items.      | <a href="#">Define Menu Items</a>       |
|                         | Respond to Menu Events. | <a href="#">Respond to Menu Events</a>  |

| Topic                     | Detail                                                | Link                                                   |
|---------------------------|-------------------------------------------------------|--------------------------------------------------------|
|                           | Handle Add-In Events.                                 | <a href="#">Handle Add-In Events</a> <sup>[2596]</sup> |
| <b>Deploy your Add-In</b> |                                                       | <a href="#">Deploy your Add-In</a> <sup>[2589]</sup>   |
|                           | Potential Pitfalls.                                   | <a href="#">Potential Pitfalls</a> <sup>[2591]</sup>   |
| <b>Manage Add-Ins</b>     |                                                       |                                                        |
|                           | Register an Add-In(developed in-house or brought-in). | <a href="#">Register an Add-In</a> <sup>[2745]</sup>   |
|                           | The Add-In Manager.                                   | <a href="#">The Add-In Manager</a> <sup>[2593]</sup>   |

### 19.3.1.1 Create Add-Ins

Before you start you must have an application development tool that is capable of creating ActiveX COM objects supporting the IDispatch interface, such as:

- Borland Delphi
- Microsoft Visual Basic
- Microsoft Visual Studio .Net

You should consider how to **define menu items**. To help with this, you could review some **examples of Automation Interfaces** - examples of code used to create Add-Ins for Enterprise Architect - on the Sparx Systems web page.

#### How to

An Enterprise Architect Add-In can be created in four steps:

| Step | Action                                                                                                                                                         | See also                                                   |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| 1    | Use a development tool to create an ActiveX COM DLL project.<br>Visual Basic users, for example, choose <i>File&gt;Create New Project-ActiveX DLL</i> .        |                                                            |
| 2    | Connect to the interface using the syntax appropriate to the language.                                                                                         | <a href="#">Connect to the Interface</a> <sup>[2382]</sup> |
| 3    | Create a COM Class and implement each of the <b>general Add-In Events</b> applicable to your Add-In. You only have to define methods for events to respond to. | <a href="#">Add-In Events</a> <sup>[2596]</sup>            |
| 4    | Add a registry key that identifies your Add-In to Enterprise Architect, as described in the <i>Deploy Add-Ins</i> topic.                                       | <a href="#">Deploy Add-Ins</a> <sup>[2589]</sup>           |

[Learn more](#)



- [Define Menu Items](#) <sup>2588</sup>
- [Examples of Automation Interfaces](#) (Online Resource)

### 19.3.1.1.1 Define Menu Items

#### Topics

| Topic                                  | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Defining Menu Items</b>             | <p>Menu items are defined by responding to the <i>GetMenuItems</i> event.</p> <p>The first time this event is called, <i>MenuName</i> is an empty string, representing the top-level menu. For a simple Add-In with just a single menu option you can return a string</p> <pre>Function EA_GetMenuItems(Repository As EA.Repository, MenuLocation As String, MenuName As String) As Variant     EA_GetMenuItems = "&amp;Joe's Add-In" End Function</pre>                                                                                                                                                                                                                                                                                  |
| <b>Defining Sub-Menus</b>              | <p>To define sub-menus, prefix a parent menu with a dash. Parent and sub-items are defined as follows:</p> <pre>Function EA_GetMenuItems(Repository As EA.Repository, MenuLocation As String, MenuName As String) As Variant     Select Case MenuName         Case ""             ' Parent Menu Item             EA_GetMenuItems = "- &amp;Joe's Add-In"         Case "- &amp;Joe's Add-In"             ' Define Sub-Menu Items using the Array notation.             ' In this example, "Diagram" and "Treeview"             compose the "Joe's Add-In" sub-menu.             EA_GetMenuItems = Array("&amp;Diagram", "&amp;Treeview")         Case Else             MsgBox "Invalid Menu", vbCritical     End Select End Function</pre> |
| <b>Defining Further Sub-Menus</b>      | <p>Similarly, you can define further sub-items:</p> <pre>Function EA_GetMenuItems(Repository As EA.Repository, MenuLocation As String, MenuName As String) As Variant     Select Case MenuName         Case ""             EA_GetMenuItems = "- Joe's Add-In"         Case "- Joe's Add-In"             EA_GetMenuItems = Array("- &amp;Diagram", "&amp;Treeview")         Case "- &amp;Diagram"             EA_GetMenuItems = "&amp;Properties"         Case Else             MsgBox "Invalid Menu", vbCritical     End Select End Function</pre>                                                                                                                                                                                        |
| <b>Enabling/Disabling menu options</b> | <p>To enable or disable menu options by default, you can use this method to show particular items to the user:</p> <pre>Sub EA_GetMenuState(Repository As EA.Repository,</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

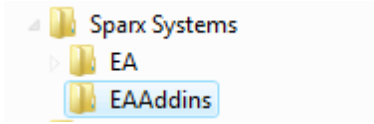
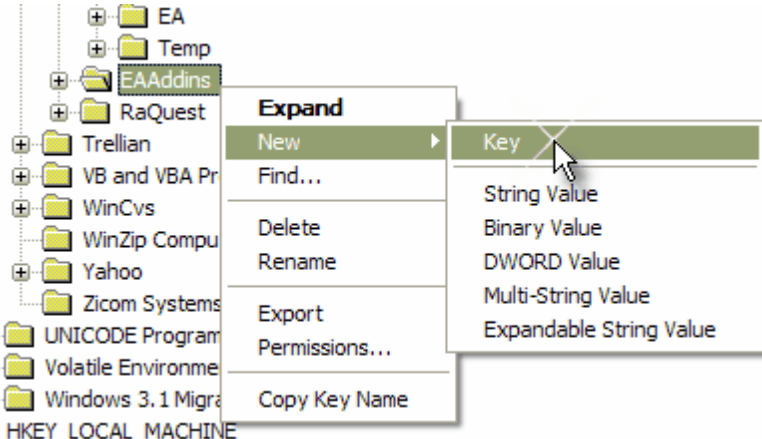
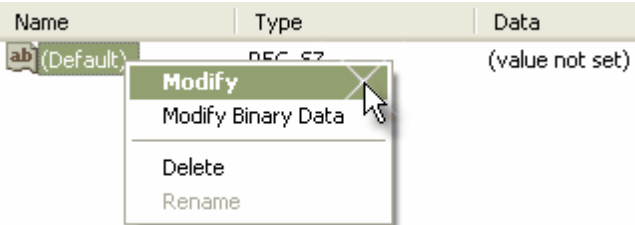
| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> Location As String, MenuName As String, ItemName As String, IsEnabled As Boolean, IsChecked As Boolean) Select Case Location Case "TreeView" ' Always enable Case "Diagram" ' Always enable Case "MainMenu" Select Case ItemName Case "&amp;Translate", "Save &amp;Project" If GetIsProjectSelected() Then IsEnabled = False End If End Select End Select IsChecked = GetIsCurrentSelection() End Sub </pre> |

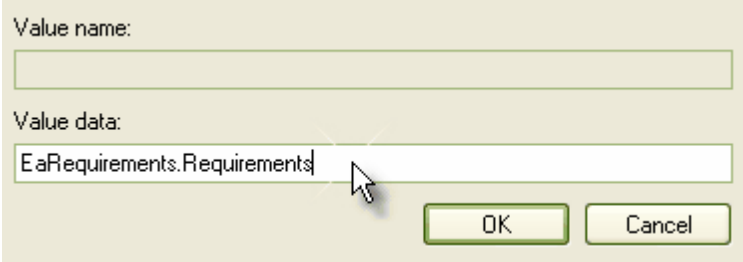
### 19.3.1.1.2 Deploy Add-Ins

#### How to

To deploy Add-Ins to users' sites

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See also |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | <p>Add the Add-In DLL file to an appropriate directory on the user's computer; that is:</p> <pre>C:\Program Files\ ( new dir )</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                  |          |
| 2    | <p>Register the DLL as appropriate to your platform:</p> <ul style="list-style-type: none"> <li>If compiled as a native Win32 DDL, such as VB6 or C++, register the DDL using the <b>regsvr32</b> command from the command prompt <pre>regsvr32 "C:\Program Files\MyCompany\EAAddin\EAAddin.dll"</pre> </li> <li>If compiled as a .NET DLL, such as C# or VB.NET, register the DLL using the <b>RegAsm</b> command from the command prompt <pre>C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\RegAsm.exe "C:\Program Files\MyCompany\EAAddin\EAAddin.dll" /codebase</pre> </li> </ul> |          |
| 3    | <p>Place a new entry into the registry using the registry editor (run <b>regedit</b>) so that Enterprise Architect recognizes the presence of your Add-In.</p>                                                                                                                                                                                                                                                                                                                                                                                                                        |          |
| 4    | <p>Add a new key value <b>EAAddIns</b> under the location:</p> <ul style="list-style-type: none"> <li><i>HKEY_CURRENT_USER\Software\Sparx Systems</i> for single users</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                     |          |

| Step | Action                                                                                                                                                                                                                                                                                                                                                      | See also |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|      | <ul style="list-style-type: none"> <li>• <code>HKEY_LOCAL_MACHINE\Software\Sparx Systems</code> for multiple users on a machine</li> </ul>                                                                                                                                 |          |
| 5    | <p>Add a new key under this key with the project name.</p>  <p>( <code>ProjectName</code> ) is not necessarily the name of your DLL, but the name of the Project; in Visual Basic, this is the value for the property <b>Name</b> corresponding to the project file.</p> |          |
| 6    | <p>Specify the default value by modifying the default value of the key.</p>                                                                                                                                                                                              |          |
| 7    | <p>Enter the value of the key by typing in the <code>( project name ) . ( class name )</code><br/> <code>EaRequirements.Requirements</code></p> <p>where <i>EaRequirements</i> is the project name, as shown in the example below:</p>                                                                                                                      |          |

| Step | Action                                                                             | See also |
|------|------------------------------------------------------------------------------------|----------|
|      |  |          |

### 19.3.1.1.3 Tricks and Traps

#### Topics

| Topic                              | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See also |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Visual Basic 5/6 Users Note</b> | <p>Visual Basic 5/6 users should note that the version number of the Enterprise Architect interface is stored in the VBP project file in a form similar to the following:</p> <pre>Reference=* \ G{ 64FB2BF4- 9EFA- 11D2- 8307- C45586000000} #2. 2#0#... \ Program Files\ Sparx Systems\ EA\ EA. TLB#Enterprise Architect Object Model 2. 02</pre> <p>If you experience problems moving from one version of Enterprise Architect to another, open the VBP file in a text editor and remove this line. Then open the project in Visual Basic and use <i>Project-References</i> to create a new reference to the Enterprise Architect Object model.</p> |          |
| <b>Add-In Fails to Load</b>        | <p>From Enterprise Architect release 7.0, Add-Ins created before 2004 are no longer supported. If an Add-In subscribes to the <code>Addn_TmpI . t l b</code> interface (2003 style), it fails on load. In this event, contact the vendor or author of the Add-In and request an upgrade.</p>                                                                                                                                                                                                                                                                                                                                                           |          |
| <b>Holding State Information</b>   | <p>It is possible for an Add-In to hold state information, meaning that data can be stored in member variables in response to one event and retrieved in another. There are some dangers in doing this:</p> <ul style="list-style-type: none"> <li>Enterprise Architect Automation Objects do not update themselves in response to user activity, to activity on other workstations, or even to the actions of other objects in the same automation client; retaining handles to such objects between calls can result in</li> </ul>                                                                                                                   |          |

| Topic                                          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also                                                                                                                                                                                            |
|------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                | <p>the second event querying objects that have no relationship with the current state of Enterprise Architect</p> <ul style="list-style-type: none"> <li>• When you close Enterprise Architect, all Add-Ins are asked to shut down; if there are any external automation clients Enterprise Architect must stay active, in which case all the Add-Ins are reloaded, losing all the data</li> <li>• Enterprise Architect acting as an automation client does not close if an Add-In still holds a reference to it (releasing all references in the <code>Disconnect()</code> event avoids this problem)</li> </ul> <p>It is recommended that unless there is a specific reason for doing so, the Add-In should use the repository parameter and its method and properties to provide the necessary data.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                     |
| <p><b>Enterprise Architect Not Closing</b></p> | <p><b>.Net Specific Issues</b></p> <p>Automation checks the use of objects and won't enable any of them to be destroyed until they are no longer being used.</p> <p>As noted in the <i>Automation Interface</i> topic, if your automation controller was written using the .NET framework, Enterprise Architect does not close even after you release all your references to it. To force the release of the COM pointers, call the memory management functions as shown below:</p> <pre>GC. Collect(); GC. Wait For Pending Finalizers();</pre> <p>Additionally, because automation clients hook into Enterprise Architect, which creates Add-Ins which in turn hook back into Enterprise Architect, it is possible to get into a deadlock situation where Enterprise Architect and the Add-Ins won't let go of one another and keep each other active. An Add-In might retain hooks into Enterprise Architect because:</p> <ul style="list-style-type: none"> <li>• It keeps a private reference to an Enterprise Architect object (see <b>Holding State Information</b> above), or</li> <li>• It has been created by .NET and the GC mechanism hasn't got around to releasing it</li> </ul> <p>There are two actions required to avoid deadlock situations:</p> <ul style="list-style-type: none"> <li>• Automation controllers must call <code>Repository.CloseAddIns()</code> at some point (presumably at the</li> </ul> | <p><a href="#">Automation Interface</a> <small>[2386]</small></p> <p><a href="#">Holding State Information</a> <small>[2591]</small></p> <p><a href="#">Add-In Events</a> <small>[2596]</small></p> |

| Topic                   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See also |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                         | <p>end of processing)</p> <ul style="list-style-type: none"> <li>• Add-Ins must release all references to Enterprise Architect in the <code>Disconnect()</code> event; see the <i>Add-In Events</i> topic for details</li> </ul> <p>It is possible that your Automation client controls a running instance of Enterprise Architect where the Add-Ins have not complied with the rule above. In this case you could call <code>Repository.Exit()</code> to terminate Enterprise Architect.</p> <p><b>Miscellaneous</b></p> <p>In developing Add-Ins using the .Net framework you must select COM Interoperability in the project's properties in order for it to be recognized as an Add-In.</p> <p>Some development environments do not automatically register COM DLLs on creation. You might have to do that manually before Enterprise Architect recognizes the Add-In.</p> <p>You can use your private Add-In key (as required for Add-In deployment) to store configuration information pertinent to your Add-In.</p> |          |
| <b>Concurrent Calls</b> | <p>In Enterprise Architect releases up to release 7.0, there is a possibility that Enterprise Architect could call two Add-In methods concurrently if the Add-In calls:</p> <ul style="list-style-type: none"> <li>• A message box</li> <li>• A modal dialog</li> <li>• VB DoEvents, .NET Application DoEvents or the equivalent in other languages</li> </ul> <p>In such cases, Enterprise Architect could initiate a second Add-In method before the first returns (re-entrancy). In release 7.0 and subsequent releases, Enterprise Architect cannot make such concurrent calls.</p> <p>If developing Add-Ins, ensure that the Add-In users are running Enterprise Architect release 7.0 or a later release to avoid any risk of concurrent method calls.</p>                                                                                                                                                                                                                                                           |          |

### 19.3.2 The Add-In Manager

[Access](#) [Extensions | Manage Add-Ins](#)

[Topics](#)



| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                     |        |                 |             |         |                                     |                   |         |                                     |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|--------|-----------------|-------------|---------|-------------------------------------|-------------------|---------|-------------------------------------|
| <b>General Usage</b> | <p>You can use the Add-In Manager to view what Add-Ins are available and to disable those not to be used.</p> <table border="1"> <thead> <tr> <th>Available Add-Ins</th> <th>Status</th> <th>Load on Startup</th> </tr> </thead> <tbody> <tr> <td>DoDAF-MODAF</td> <td>Enabled</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Zachman Framework</td> <td>Enabled</td> <td><input checked="" type="checkbox"/></td> </tr> </tbody> </table> <p>To enable an Add-In for use, select its <b>Load on Startup</b> check box. To disable an Add-In, deselect the checkbox.</p> | Available Add-Ins                   | Status | Load on Startup | DoDAF-MODAF | Enabled | <input checked="" type="checkbox"/> | Zachman Framework | Enabled | <input checked="" type="checkbox"/> |
| Available Add-Ins    | Status                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Load on Startup                     |        |                 |             |         |                                     |                   |         |                                     |
| DoDAF-MODAF          | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | <input checked="" type="checkbox"/> |        |                 |             |         |                                     |                   |         |                                     |
| Zachman Framework    | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | <input checked="" type="checkbox"/> |        |                 |             |         |                                     |                   |         |                                     |

#### Notes

- Enterprise Architect must be restarted for changes to take effect

### 19.3.3 Add-In Search

#### Topics

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See also  |             |            |                                |                   |                                                      |                   |                                                                                                     |                                                                                                                                                                                                                                                           |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|-------------|------------|--------------------------------|-------------------|------------------------------------------------------|-------------------|-----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>General Usage</b> | <p>Enterprise Architect enables Extensions to integrate with the Model Search. Searches can be defined that execute a method within your Add-In and display your results in an integrated way.</p> <p>The method that runs the search must be structured in the following way:</p> <pre>variant &lt;method name&gt; (Rep as Repository, SearchText as String, XMLResults as String)</pre> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>Rep</b></td> <td>The currently open repository.</td> </tr> <tr> <td><b>SearchText</b></td> <td>An optional field that you can fill in through the M</td> </tr> <tr> <td><b>XMLResults</b></td> <td>At completion of the method, this should contain<br/>should be an XML string that conforms to the Se</td> </tr> </tbody> </table> | Parameter | Description | <b>Rep</b> | The currently open repository. | <b>SearchText</b> | An optional field that you can fill in through the M | <b>XMLResults</b> | At completion of the method, this should contain<br>should be an XML string that conforms to the Se | <p><a href="#">Model Search</a> </p> <p><a href="#">Search Data Format</a> </p> |
| Parameter            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |           |             |            |                                |                   |                                                      |                   |                                                                                                     |                                                                                                                                                                                                                                                           |
| <b>Rep</b>           | The currently open repository.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |           |             |            |                                |                   |                                                      |                   |                                                                                                     |                                                                                                                                                                                                                                                           |
| <b>SearchText</b>    | An optional field that you can fill in through the M                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |           |             |            |                                |                   |                                                      |                   |                                                                                                     |                                                                                                                                                                                                                                                           |
| <b>XMLResults</b>    | At completion of the method, this should contain<br>should be an XML string that conforms to the Se                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |           |             |            |                                |                   |                                                      |                   |                                                                                                     |                                                                                                                                                                                                                                                           |

| Topic                 | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also  |             |  |  |  |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|-------------|--|--|--|
|                       | <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Parameter | Description |  |  |  |
| Parameter             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |           |             |  |  |  |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |           |             |  |  |  |
| <b>Return</b>         | The method must return a value for the results to be displayed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |           |             |  |  |  |
| <b>Advanced Usage</b> | <p>In addition to the displayed results, two additional hidden fields can be passed into the XML that provide special functionality.</p> <p><b>CLASSTYPE</b></p> <p>Returning a field of CLASSTYPE, containing the <i>Object_Type</i> value from the <i>t_object</i> table, displays the appropriate icon in the column you place the field.</p> <p><b>CLASSGUID</b></p> <p>Returning a field of CLASSGUID, containing an <i>ea_guid</i> value, enables the Model Search to track the object in the Project Browser and open the Properties window for the element by double-clicking in the Model Search.</p> |           |             |  |  |  |

### 19.3.3.1 XML Format (Search Data)

#### Topics

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See also                             |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| <b>General Usage</b> | The XML below provides the format for the <i>sSearchData</i> parameter of the <i>RunModelSearch</i> method.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | <a href="#">Repository</a><br>[2466] |
| <b>Example Code</b>  | <pre>&lt;ReportViewData UID="\ MySearchID" &gt;   &lt;!--     //The UID attribute enables XML type searches     to persist column information. That is, if you     run the search, group by column or adjust column     widths, then close the window and run the search     again, the format/organization changes are     retained.To avoid persisting column     arrangements, leave the attribute value blank or     remove it altogether.     // Use this section to declare all possible     fields - columns that appear in Enterprise     Architect's search window - that are used below     in &lt;Rows/&gt;.     // The order of the columns of information to     be appended here must match the order that the     search run in Enterprise Architect would     normally display.     // Furthermore, if you append results onto a     custom SQL Search, then the order used in your     Custom SQL must match the order used below.</pre> |                                      |



| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <pre> -- &gt;  &lt;Fields&gt;   &lt;Field name="" /&gt;   &lt;Field name="" /&gt;   &lt;Field name="" /&gt;   &lt;Field name="" /&gt; &lt;/Fields&gt;  &lt;Rows&gt;   &lt;Row&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;   &lt;/Row&gt;   &lt;Row&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;   &lt;/Row&gt;   &lt;Row&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;   &lt;/Row&gt; &lt;/Rows&gt; &lt;/ReportViewData&gt; </pre> |          |

### 19.3.4 Add-In Events

All Enterprise Architect Add-Ins can choose to respond to the following general Add-In events:

| Topic                                                                                                                                                                 | Link                                              |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| <i>EA_Connect</i> - enables Add-Ins to identify their type and to respond to Enterprise Architect start up.                                                           | <a href="#">EA_Connect</a> <sup>[2597]</sup>      |
| <i>EA_Disconnect</i> - enables the Add-In to respond to user requests to disconnect the model branch from an external project.                                        | <a href="#">EA_Disconnect</a> <sup>[2598]</sup>   |
| <i>EA_GetMenuItems</i> - enables the Add-In to provide the Enterprise Architect user interface with additional Add-In menu options in various context and main menus. | <a href="#">EA_GetMenuItems</a> <sup>[2598]</sup> |
| <i>EA_MenuClick</i> - received by an Add-In in response to user selection of a menu option.                                                                           | <a href="#">EA_MenuClick</a> <sup>[2601]</sup>    |
| <i>EA_GetMenuState</i> - enables the Add-In to set a particular menu option to either enabled or disabled.                                                            | <a href="#">EA_GetMenuState</a> <sup>[2599]</sup> |

| Topic                                                                                                                                                          | Link                                                         |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| <i>EA_ShowHelp</i> - enables the Add-In to show a help topic for a particular menu option.                                                                     | <a href="#">EA_ShowHelp</a> <sup>[2604]</sup>                |
| <i>EA_OnOutputItemClicked</i> - informs Add-Ins that the user has clicked on a list entry in the system tab or one of the user defined output tabs.            | <a href="#">EA_OnOutputItemClicked</a> <sup>[2602]</sup>     |
| <i>EA_OnOutputItemDoubleClicked</i> - informs Add-Ins that the user has used the mouse to double-click on a list entry in one of the user-defined output tabs. | <a href="#">EA_OnOutputItemDoubleClick</a> <sup>[2603]</sup> |

### 19.3.4.1 EA\_Connect

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See also                                                                                                 |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_Connect</i> events enable Add-Ins to identify their type and to respond to Enterprise Architect start up.</p> <p>This event occurs when Enterprise Architect first loads your Add-In. Enterprise Architect itself is loading at this time so that while a Repository object is supplied, there is limited information that you can extract from it.</p> <p>The chief uses for <i>EA_Connect</i> are in initializing global Add-In data and for identifying the Add-In as an MDG Add-In.</p> | <p><a href="#">MDG Add-In</a><sup>[2671]</sup></p> <p><a href="#">EA Disconnect</a><sup>[2598]</sup></p> |

#### Syntax:

#### Function *EA\_Connect(Repository As EA.Repository) As String*

The *EA\_Connect* function syntax has the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                        |
|-------------------|-------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An EA.Repository object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

#### Return Value

A string identifying a specialized type of Add-In:

| Type  | Details                                                                                  |
|-------|------------------------------------------------------------------------------------------|
| "MDG" | MDG Add-Ins receive <a href="#">MDG Events</a> <sup>[2672]</sup> and extra menu options. |
| ""    | A none-specialized Add-In.                                                               |

### 19.3.4.2 EA\_Disconnect

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See also                                     |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| <b>Details</b> | <p>The <i>EA_Disconnect</i> event enables the Add-In to respond to user requests to disconnect the model branch from an external project.</p> <p>This function is called when the Enterprise Architect closes. If you have stored references to Enterprise Architect objects (not particularly recommended anyway), you must release them here.</p> <p>In addition, .NET users must call memory management functions as shown below:</p> <pre>GC. Collect ( ) ; GC. Wait For PendingFinalizers ( ) ;</pre> | <a href="#">EA_Connect</a> <sup>[2597]</sup> |

#### Syntax:

*Sub EA\_Disconnect()*

#### Return Value:

None.

### 19.3.4.3 EA\_GetMenuItems

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See also                                                                                                                                                   |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p>The <i>EA_GetMenuItems</i> event enables the Add-In to provide the Enterprise Architect user interface with additional Add-In menu options in various context and main menus. When a user selects an Add-In menu option, an event is raised and passed back to the Add-In that originally defined that menu option.</p> <p>This event is raised just before Enterprise Architect has to show particular menu options to the user, and its use is</p> | <a href="#">Define Menu Items</a> <sup>[2588]</sup><br><a href="#">EA_MenuClick</a> <sup>[2601]</sup><br><a href="#">EA_GetMenuState</a> <sup>[2595]</sup> |

| Topic | Detail                                           | See also |
|-------|--------------------------------------------------|----------|
|       | described in the <i>Define Menu Items</i> topic. |          |

**Syntax:**

**Function** *EA\_GetMenuItems*(*Repository As EA.Repository, MenuLocation As String, MenuName As String*) *As Variant*

The *EA\_GetMenuItems* function syntax has the following elements:

| Parameter           | Type                                    | Direction | Description                                                                                                                                                        |
|---------------------|-----------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MenuLocation</b> | <i>String</i>                           |           | A string representing the part of the user interface that brought up the menu. This can be <i>TreeView</i> , <i>MainMenu</i> or <i>Diagram</i> .                   |
| <b>MenuName</b>     | <i>String</i>                           |           | The name of the parent menu for which sub-items are to be defined. In the case of the top-level menu this is an empty string.                                      |
| <b>Repository</b>   | <a href="#">EA.Repository</a><br>[2448] | IN        | An EA.Repository object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

**Return Value:**

One of the following types:

- A string indicating the label for a single menu option
- An array of strings indicating a multiple menu options
- Empty (Visual Basic/VB.NET) or null (C#) to indicate that no menu should be displayed

In the case of the top-level menu it should be a single string or an array containing only one item, or empty/null.

**19.3.4.4 EA\_GetMenuState****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                            | See also                                                                                 |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| <b>Details</b> | The <i>EA_GetMenuState</i> event enables the Add-In to set a particular menu option to either enabled or disabled. This is useful when dealing with locked packages and other situations where it is convenient to show a menu option, but not enable it for use. | <a href="#">Define Menu Items</a><br>[2588]<br><a href="#">EA_GetMenuItems</a><br>[2598] |

| Topic | Detail                                                                                                                                                             | See also |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | This event is raised just before Enterprise Architect has to show particular menu options to the user. Its use is described in the <i>Define Menu Items</i> topic. |          |

**Syntax:**

***Sub EA\_GetMenuState(Repository as EA.Repository, MenuLocation As String, MenuName as String, ItemName as String, IsEnabled as Boolean, IsChecked as Boolean)***

The *EA\_GetMenuState* function syntax has the following elements:

| Parameter           | Type                                    | Direction | Description                                                                                                                                                               |
|---------------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>IsChecked</b>    | <i>Boolean</i>                          |           | Set to <b>True</b> to check this particular menu option.                                                                                                                  |
| <b>IsEnabled</b>    | <i>Boolean</i>                          |           | Set to <b>False</b> to disable this particular menu option.                                                                                                               |
| <b>ItemName</b>     | <i>String</i>                           |           | The name of the option actually clicked; for example, <i>Create a New Invoice</i> .                                                                                       |
| <b>MenuLocation</b> | <i>String</i>                           |           | A string representing the part of the user interface that brought up the menu. This can be <i>TreeView</i> , <i>MainMenu</i> or <i>Diagram</i> .                          |
| <b>MenuName</b>     | <i>String</i>                           |           | The name of the parent menu for which sub-items must be defined.<br><br>In the case of the top-level menu it is an empty string.                                          |
| <b>Repository</b>   | <a href="#">EA.Repository</a><br>[2448] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

**Return Value:**

None.

### 19.3.4.5 EA\_MenuClick

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also                                                                                                   |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_MenuClick</i> events are received by an Add-In in response to user selection of a menu option.</p> <p>The event is raised when the user clicks on a particular menu option. When a user clicks on one of your non-parent menu options, your Add-In receives a <i>MenuClick</i> event, defined as follows:</p> <pre>Sub EA_MenuClick( Repository As EA. Repository, ByVal MenuName As String, ByVal ItemName As String)</pre> <p>The code below illustrates an example of use:</p> <pre>    If MenuName = "- &amp;Diagram" And ItemName = "&amp;Properties" then         MsgBox Repository. Get Current Diagram Name, vbInformation     Else         MsgBox "Not Implemented", vbCritical     End If</pre> <p>Notice that your code can directly access Enterprise Architect data and UI elements using <b>Repository</b> methods.</p> | <p><a href="#">Repository</a><sup>[244B]</sup></p> <p><a href="#">EA_GetMenuItems</a><sup>[259B]</sup></p> |

#### Syntax:

***Sub EA\_MenuClick(Repository As EA.Repository, MenuLocation As String, MenuName As String, ItemName As String)***

The *EA\_GetMenuClick* function syntax has the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                               |
|-------------------|-------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ItemName</b>   | <i>String</i>                                   |           | The name of the option actually clicked, for example, <i>Create a New Invoice</i> .                                                                                       |
| <b>MenuName</b>   | <i>String</i>                                   |           | The name of the parent menu for which sub-items are to be defined.<br><br>In the case of the top-level menu this is an empty string.                                      |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[244B]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

**Return Value:**

None.

**19.3.4.6 EA\_OnOutputItemClicked****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See also                                                       |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnOutputItemClicked</i> events inform Add-Ins that the user has clicked on a list entry in the system tab or one of the user defined output tabs.</p> <p>Usually an Add-In responds to this event in order to capture activity on an output tab they had previously created through a call to <i>Repository.AddTab()</i>.</p> <p>Note that every loaded Add-In receives this event for every click on an output tab in Enterprise Architect, irrespective of whether the Add-In created that tab. Add-Ins should therefore check the <i>TabName</i> parameter supplied by this event to ensure that they are not responding to other Add-Ins' events.</p> | <a href="#">EA_OnOutputItemDoubleClicked</a> <sup>[2603]</sup> |

**Syntax:**

**EA\_OnOutputItemClicked(*Repository As EA.Repository, TabName As String, LineText As String, ID As Long*)**

The *EA\_OnOutputItemClicked* function syntax has the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                               |
|-------------------|-------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ID</b>         | <i>Long</i>                                     | IN        | The ID value specified in the original call to <i>Repository.WriteOutput()</i> .                                                                                          |
| <b>LineText</b>   | <i>String</i>                                   | IN        | The text that had been supplied as the String parameter in the original call to <i>Repository.WriteOutput()</i> .                                                         |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| <b>TabName</b>    | <i>String</i>                                   | IN        | The name of the tab that the click occurred in. Usually this would have been created through <i>Repository.AddTab()</i> .                                                 |

**Return Value:**

None.

### 19.3.4.7 EA\_OnOutputItemDoubleClicked

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See also                                         |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnOutputItemDoubleClicked</i> events informs Add-Ins that the user has used the mouse to double-click on a list entry in one of the user-defined output tabs.</p> <p>Usually an Add-In responds to this event in order to capture activity on an output tab they had previously created through a call to <i>Repository.AddTab()</i>.</p> <p>Note that every loaded Add-In receives this event for every double-click on an output tab in Enterprise Architect, irrespective of whether the Add-In created that tab; Add-Ins should therefore check the <i>TabName</i> parameter supplied by this event to ensure that they are not responding to other Add-Ins' events.</p> | <a href="#">EA_OnOutputItemClicked</a><br>[2602] |

#### Syntax:

**EA\_OnOutputItemDoubleClicked(Repository As EA.Repository, TabName As String, LineText As String, ID As Long)**

The *EA\_OnOutputItemClicked* function syntax contains the following elements:

| Parameter         | Type                                    | Direction | Description                                                                                                                                                               |
|-------------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ID</b>         | <i>Long</i>                             | IN        | The ID value specified in the original call to <i>Repository.WriteOutput()</i> .                                                                                          |
| <b>LineText</b>   | <i>String</i>                           | IN        | The text that had been supplied as the String parameter in the original call to <i>Repository.WriteOutput()</i> .                                                         |
| <b>Repository</b> | <a href="#">EA.Repository</a><br>[2448] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model; poll its members to retrieve model data and user interface status information. |
| <b>TabName</b>    | <i>String</i>                           | IN        | The name of the tab that the click occurred in; usually this would have been created through <i>Repository.AddTab()</i> .                                                 |

#### Return Value:

None.



### 19.3.4.8 EA\_ShowHelp

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                    | See also                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| <b>Details</b> | <p>The <i>EA_ShowHelp</i> event enables the Add-In to show a help topic for a particular menu option. When the user has an Add-In menu option selected, pressing ( <b>F1</b> ) can be related to the required Help topic by the Add-In and a suitable help message shown.</p> <p>This event is raised when the user presses ( <b>F1</b> ) on a menu option that is not a parent menu.</p> | <a href="#">EA_GetMenuItem</a><br><a href="#">s [2598]</a> |

#### Syntax:

**Sub EA\_ShowHelp(Repository as EA.Repository, MenuLocation As String, MenuName as String, ItemName as String)**

The *EA\_ShowHelp* function syntax contains the following elements:

| Parameter           | Type                                                    | Direction | Description                                                                                                                                                               |
|---------------------|---------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ItemName</b>     | <i>String</i>                                           |           | The name of the option actually clicked, for example, <b>Create a New Invoice</b> .                                                                                       |
| <b>MenuLocation</b> | <i>String</i>                                           |           | A string representing the part of the user interface that brought up the menu. This can be <i>Treeview</i> , <i>MainMenu</i> or <i>Diagram</i> .                          |
| <b>MenuName</b>     | <i>String</i>                                           |           | The name of the parent menu for which sub-items are to be defined. In the case of the top-level menu this is an empty string.                                             |
| <b>Repository</b>   | <a href="#">EA.Repository</a><br><a href="#">[2448]</a> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

#### Return Value:

None.

### 19.3.5 Broadcast Events

#### Overview:

The following general Broadcast events are sent to all loaded Add-Ins. For an Add-In to receive the event, they must first implement the required automation event interface. If Enterprise Architect detects that the Add-

If it has the required interface, the event is dispatched to the Add-In.

| Topic                                  | Link                                                                                                                                 |
|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Add-In Licence Management Events       | <a href="#">Add-In Licence Management Events</a>  |
| Compartment Events                     | <a href="#">Compartment Events</a>                |
| Context Item Events                    | <a href="#">Context Item Events</a>               |
| File Close Event                       | <a href="#">File Close Event</a>                  |
| File New Event                         | <a href="#">File New Event</a>                    |
| File Open Event                        | <a href="#">File Open Event</a>                   |
| Model Validation Broadcasts            | <a href="#">Model Validation Broadcasts</a>      |
| Post Close Diagram Event               | <a href="#">Post Close Diagram Event</a>        |
| Post Initialization Event              | <a href="#">EA_OnPostInitialized</a>            |
| Post New Events                        | <a href="#">Post-New Events</a>                 |
| Post Open Diagram Event                | <a href="#">Post Open Diagram Event</a>         |
| Pre-Deletion Events                    | <a href="#">Pre-Deletion Events</a>             |
| Pre-Exit Instance (not currently used) | <a href="#">PreExit Instance</a>                |
| Pre-New Events                         | <a href="#">Pre-New Events</a>                  |
| Retrieve Model Template Event          | <a href="#">Retrieve Model Template Event</a>   |
| Tagged Value Broadcasts                | <a href="#">Tagged Value Broadcasts</a>         |
| Technology Events                      | <a href="#">Technology Events</a>               |

| Topic                | Link                                                   |
|----------------------|--------------------------------------------------------|
| Transformation Event | <a href="#">Transformation Event</a> <sup>[2619]</sup> |

**MDG Events** add quite a number of additional events, but the Add-In must first have registered as an MDG-style Add-In, rather than as a generic Add-In.

#### Learn more

- [MDG Events](#) <sup>[2672]</sup>

### 19.3.5.1 Add-In License Management Events

Enterprise Architect Add-Ins can respond to the following events associated with Add-In License Management:

| Topic                         | Link                                                            |
|-------------------------------|-----------------------------------------------------------------|
| EA_AddinLicenseValidate       | <a href="#">EA_AddinLicenseValidate</a> <sup>[2606]</sup>       |
| EA_AddinLicenseGetDescription | <a href="#">EA_AddinLicenseGetDescription</a> <sup>[2607]</sup> |

#### Learn more

- [Register Add-In](#) <sup>[2745]</sup>

#### 19.3.5.1.1 EA\_AddinLicenseValidate

##### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See also                                                        |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_AddinLicenseValidate</i> is broadcast to all Add-Ins, providing them with a chance to use the Add-In key to determine the level of functionality to provide. The event is broadcast in response to a non-standard Enterprise Architect key being entered into the License Management dialog.</p> <p>For the Add-In to validate itself against this key, the Add-In's <i>EA_AddinLicenseValidate</i> handler should return <b>true</b> to advise Enterprise Architect that the license has been validated. The <i>EA_AddinLicenseValidate</i> event is broadcast to all Add-Ins, therefore one license can validate many Add-Ins.</p> <p>If an Add-In elects to handle a license key by returning true to <i>EA_AddinLicenseValidate</i>, it is called upon to provide a description of the license key through the</p> | <a href="#">EA_AddinLicenseGetDescription</a> <sup>[2607]</sup> |

| Topic | Detail                                                                                                                                                                                                                      | See also |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <i>EA_AddinLicenseGetDescription</i> event. If more than one Add-In elects to handle a license key, the first Add-In that returns <b>true</b> to <i>EA_AddinLicenseValidate</i> is queried for the license key description. |          |

**Syntax:**

**Function** *EA\_AddinLicenseValidate*(*Repository As EA.Repository, AddinKey As String*) **As Boolean**

The *EA\_AddinLicenseValidate* function syntax contains the following elements.

| Parameter         | Type                                            | Direction | Description                                                                                                                                                        |
|-------------------|-------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AddinKey</b>   | <i>String</i>                                   | IN        | The Add-in license key that has been entered in Enterprise Architect's License Management dialog.                                                                  |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An EA.Repository object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

**Return Value:**

For the Add-in to validate against this key it should return **true** to indicate to Enterprise Architect that the key is valid and has been handled.

**19.3.5.1.2 EA\_AddinLicenseGetDescription****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                  | See also |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | Before Enterprise Architect's License Management dialog is displayed, <i>EA_AddinLicenseGetDescription</i> is sent once for each Add-In key to the first Add-In that elected to handle that key.<br><br>The value returned by <i>EA_AddinLicenseGetDescription</i> is used by Enterprise Architect as the key's plain text description. |          |

**Syntax:**

**Function** *EA\_AddinLicenseGetDescription* (*Repository as EA.Repository, AddinKey as String*) **As String**

The *EA\_OnPostInitialized* function syntax contains the following elements.

| Parameter         | Type                                            | Direction | Description                                                                                                                                                        |
|-------------------|-------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AddinKey</b>   | <i>String</i>                                   | IN        | The Add-In license key that Enterprise Architect requires a description for.                                                                                       |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An EA.Repository object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

**Return Value:**

A String containing a plain text description of the provided AddinKey.

**19.3.5.2 Compartment Events**

Enterprise Architect Add-Ins can respond to the following events associated with user-generated element compartments:

| Topic                         | Link                                                            |
|-------------------------------|-----------------------------------------------------------------|
| EA_QueryAvailableCompartments | <a href="#">EA_QueryAvailableCompartments</a> <sup>[2608]</sup> |
| EA_GetCompartmentData         | <a href="#">EA_GetCompartmentData</a> <sup>[2609]</sup>         |

**19.3.5.2.1 EA\_QueryAvailableCompartments****Topics**


| Topic          | Detail                                                                                                                                                                                                                                                                                       | See also                                                |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| <b>Details</b> | <p>This event occurs when Enterprise Architect's diagrams are refreshed. It is a request for the Add-In to provide a list of user-defined compartments.</p> <p>The <i>EA_GetCompartmentData</i> event then queries each object for the data to display in each user-defined compartment.</p> | <a href="#">EA_GetCompartmentData</a> <sup>[2609]</sup> |

**Syntax:**

**Function** *EA\_QueryAvailableCompartments(Repository As EA.Repository) As Variant*

The *EA\_QueryAvailableCompartments* function syntax contains the following elements:

| Parameter         | Type                          | Direction | Description                                                    |
|-------------------|-------------------------------|-----------|----------------------------------------------------------------|
| <b>Repository</b> | <a href="#">EA.Repository</a> | IN        | An <i>EA.Repository</i> object representing the currently open |

| Parameter | Type                                                                              | Direction | Description                                                                                                |
|-----------|-----------------------------------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------|
|           |  |           | Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

**Return Value:**

A *String* containing a comma-separated list of user-defined compartments.

**Example**

```

Function EA_QueryAvailableCompartments(Repository As EA.Repository) As Variant
 Dim sReturn As String
 sReturn = ""
 If m_FirstCompartmentVisible = True Then
 sReturn = sReturn + "first,"
 End If
 If m_SecondCompartmentVisible = True Then
 sReturn = sReturn + "second,"
 End If
 If m_ThirdCompartmentVisible = True Then
 sReturn = sReturn + "third,"
 End If

 If Len(sReturn) > 0 Then
 sReturn = Left(sReturn, Len(sReturn) - 1)
 End If

 EA_QueryAvailableCompartments = sReturn
End Function

```

**19.3.5.2.2 EA\_GetCompartmentData****Topics**

| Topic          | Detail                                                                                                                                                              | See also |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | This event occurs when Enterprise Architect is instructed to redraw an element. It requests that the Add-In provide the data to populate the element's compartment. |          |

**Syntax:**

**Function** EA\_GetCompartmentData(*Repository* As EA.Repository, *sCompartment* As String, *sGUID* As String, *oType* As EA.ObjectType) As Variant

The *EA\_QueryAvailableCompartments* function syntax contains the following elements:

| Parameter           | Type                                                   | Direction | Description                                                                                                                                                               |
|---------------------|--------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>oType</b>        | <i>ObjectType</i>                                      | IN        | The type of the element for which data is being requested.                                                                                                                |
| <b>Repository</b>   | <a href="#">EA.Repository</a><br><small>[2448]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| <b>sCompartment</b> | <i>String</i>                                          | IN        | The name of the compartment for which data is being requested.                                                                                                            |
| <b>sGUID</b>        | <i>String</i>                                          | IN        | The GUID of the element for which data is being requested.                                                                                                                |

**Return Value:**

A variant containing a formatted string. See the example below to view the format.

**Example**

```
Function EA_GetCompartmentData(Repository As EA.Repository, sCompartment
As String, sGUID As String, oType As EA.ObjectType) As Variant
```

```
 If Repository Is Nothing Then
 Exit Function
 End If

 Dim sCompartmentData As String
 Dim oXML As MSXML2.DOMDocument
 Dim Nodes As MSXML2.IXMLDOMNodeList
 Dim Node1 As MSXML2.IXMLDOMNode
 Dim Node As MSXML2.IXMLDOMNode
 Dim sData As String

 sCompartmentData = ""
 Set oXML = New MSXML2.DOMDocument
 sData = ""

 On Error GoTo ERR_GetCompartmentData

 oXML.LoadXML (Repository.GetTreeXMLByGUID(sGUID))
 Set Node1 = oXML.selectSingleNode("// ModelItem")

 If Node1 Is Nothing Then
 Exit Function
 End If

 sCompartmentData = sCompartmentData + " Name=" + sCompartment + ";"
 sCompartmentData = sCompartmentData + " Owner GUID=" + sGUID + ";"
 sCompartmentData = sCompartmentData +
" Options=SkipIfOnDiagram&_eq_^1&_sc_^"

 Select Case sCompartment
 Case "parts"
 Set Nodes = Node1.selectNodes(" ModelItem(@Metatype="" Part "") ")
 For Each Node In Nodes
```

```

 sData = sData + "Data&_eq_^" + Node.Attributes.GetNamedItem
("Name").nodeValue + "&_sc_^"
 sData = sData + "GUID&_eq_^" + Node.Attributes.GetNamedItem
("GUID").nodeValue + "&_sc_^,"
 Next

 Case "ports"
 Set Nodes = Node1.selectNodes("ModelItem(@Metatype="Port") ")
 For Each Node In Nodes
 sData = sData + "Data&_eq_^" + Node.Attributes.GetNamedItem
("Name").nodeValue + "&_sc_^"
 sData = sData + "GUID&_eq_^" + Node.Attributes.GetNamedItem
("GUID").nodeValue + "&_sc_^,"
 Next

 End Select

' If there's no data to display, then don't return any compartment
data
If sData <> "" Then
 sCompartmentData = sCompartmentData + "CompartmentData=" + sData +
";"
Else
 sCompartmentData = ""
End If

EA_GetCompartmentData = sCompartmentData
Exit Function

ERR_GetCompartmentData:
EA_GetCompartmentData = ""

End Function

```

### 19.3.5.3 Context Item Events

Enterprise Architect Add-Ins can respond to the following events associated with changing context:

| Topic                          | Link                                                                       |
|--------------------------------|----------------------------------------------------------------------------|
| EA_OnContextItemChanged        | <a href="#">EA_OnContextItemChanged</a><br>[2612]                          |
| EA_OnContextItemDoubleClicked  | <a href="#">EA_OnContextItemDoubleClic</a><br><a href="#">ked</a> [2613]   |
| EA_OnNotifyContextItemModified | <a href="#">EA_OnNotifyContextItem Modifi</a><br><a href="#">ed</a> [2614] |



### 19.3.5.3.1 EA\_OnContextItemChanged

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                               | See also                                                                                                                                                                                         |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnContextItemChanged</i> notifies Add-Ins that a different item is now in context.</p> <p>This event occurs after a user has selected an item anywhere in the Enterprise Architect GUI. Add-Ins that require knowledge of the current item in context can subscribe to this broadcast function. If <b>ot = otRepository</b>, then this function behaves the same as <i>EA_FileOpen</i>.</p> | <p><a href="#">EA_FileOpen</a><sup>[2616]</sup></p> <p><a href="#">EA_OnContextItemDoubleClicked</a><sup>[2613]</sup></p> <p><a href="#">EA_OnNotifyContextItemModified</a><sup>[2614]</sup></p> |

#### Syntax:

**Sub EA\_OnContextItemChanged(Repository As EA.Repository, GUID As String, ot as EA.ObjectType)**

The *EA\_OnContextItemChanged* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------|-------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GUID</b>       | <i>String</i>                                   | IN        | <p>Contains the GUID of the new context item. This value corresponds to the following properties, depending on the value of the <b>ot</b> parameter:</p> <p><i>ot (ObjectType)</i> - GUID value</p> <p><i>otElement</i> - <i>Element.ElementGUID</i></p> <p><i>otPackage</i> - <i>Package.PackageGUID</i></p> <p><i>otDiagram</i> - <i>Diagram.DiagramGUID</i></p> <p><i>otAttribute</i> - <i>Attribute.AttributeGUID</i></p> <p><i>otMethod</i> - <i>Method.MethodGUID</i></p> <p><i>otConnector</i> - <i>Connector.ConnectorGUID</i></p> <p><i>otRepository</i> - NOT APPLICABLE, GUID is an empty string</p> |
| <b>ot</b>         | <i>EA.ObjectType</i>                            | IN        | Specifies the type of the new context item.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                                                                                                                                                                                                                                                                                                                                                                                                       |

#### Return Value:

None.

### 19.3.5.3.2 EA\_OnContextItemDoubleClicked

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                        | See also                                                                                                                                 |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnContextItemDoubleClicked</i> notifies Add-Ins that the user has double-clicked the item currently in context.</p> <p>This event occurs when a user has double-clicked (or pressed ( <b>Enter</b> ) ) on the item in context, either in a diagram or in the Project Browser. Add-Ins to handle events can subscribe to this broadcast function.</p> | <p><a href="#">EA_OnContextItemChanged</a> <sup>[2612]</sup></p> <p><a href="#">EA_OnNotifyContextItemModified</a> <sup>[2614]</sup></p> |

#### Syntax:

**Function** EA\_OnContextItemDoubleClicked(*Repository As EA.Repository, GUID As String, ot as EA.ObjectType*)

The *EA\_OnContextItemDoubleClicked* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------|-------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GUID</b>       | <i>String</i>                                   | IN        | <p>Contains the GUID of the new context item. This value corresponds to the following properties, depending on the value of the <b>ot</b> parameter:</p> <p><i>ot (ObjectType)</i> - <i>GUID value</i></p> <p><i>otElement</i> - <i>Element.ElementGUID</i></p> <p><i>otPackage</i> - <i>Package.PackageGUID</i></p> <p><i>otDiagram</i> - <i>Diagram.DiagramGUID</i></p> <p><i>otAttribute</i> - <i>Attribute.AttributeGUID</i></p> <p><i>otMethod</i> - <i>Method.MethodGUID</i></p> <p><i>otConnector</i> - <i>Connector.ConnectorGUID</i></p> |
| <b>ot</b>         | <i>EA.ObjectType</i>                            | IN        | Specifies the type of the new context item.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                                                                                                                                                                                                                                                                                                                                         |

#### Return Value:

Return **True** to notify Enterprise Architect that the double-click event has been handled by an Add-In.

Return **False** to enable Enterprise Architect to continue processing the event.

### 19.3.5.3.3 EA\_OnNotifyContextItemModified

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                        | See also                                                                                                                                |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnNotifyContextItemModified</i> notifies Add-Ins that the current context item has been modified.</p> <p>This event occurs when a user has modified the context item. Add-Ins that require knowledge of when an item has been modified can subscribe to this broadcast function.</p> | <p><a href="#">EA_OnContextItemChanged</a> <sup>[2612]</sup></p> <p><a href="#">EA_OnContextItemDoubleClicked</a> <sup>[2613]</sup></p> |

#### Syntax:

**Sub EA\_OnNotifyContextItemModified(Repository As EA.Repository, GUID As String, ot as EA.ObjectType)**

The *EA\_OnNotifyContextItemModified* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------|-------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GUID</b>       | <i>String</i>                                   | IN        | <p>Contains the GUID of the new context item. This value corresponds to the following properties, depending on the value of the <b>ot</b> parameter:</p> <ul style="list-style-type: none"> <li>• <i>ot(ObjectType)</i> - <i>GUID value</i></li> <li>• <i>otElement</i> - <i>Element.ElementGUID</i></li> <li>• <i>otPackage</i> - <i>Package.PackageGUID</i></li> <li>• <i>otDiagram</i> - <i>Diagram.DiagramGUID</i></li> <li>• <i>otAttribute</i> - <i>Attribute.AttributeGUID</i></li> <li>• <i>otMethod</i> - <i>Method.MethodGUID</i></li> <li>• <i>otConnector</i> - <i>Connector.ConnectorGUID</i></li> </ul> |
| <b>ot</b>         | <i>EA.ObjectType</i>                            | IN        | Specifies the type of the new context item.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                                                                                                                                                                                                                                                                                                                                                                                                             |

#### Return Value:

None.

### 19.3.5.4 EA\_FileClose

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                              | See also                                                                          |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| <b>Details</b> | <p>The <i>EA_FileClose</i> event enables the Add-In to respond to a File Close event. When Enterprise Architect closes an opened Model file, this event is raised and passed to all Add-Ins implementing this method.</p> <p>This event occurs when the model currently opened within Enterprise Architect is about to be closed (when another model is about to be opened or when Enterprise Architect is about to shutdown)..</p> | <a href="#">EA_FileOpen</a><br>[2615]<br><br><a href="#">EA_FileNew</a><br>[2615] |

#### Syntax:

##### *Sub EA\_FileClose(Repository As EA.Repository)*

The *EA\_FileClose* function syntax contains the following elements:

| Parameter         | Type                                    | Direction | Description                                                                                                                                                                                 |
|-------------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b> | <a href="#">EA.Repository</a><br>[2448] | IN        | <p>An <i>EA.Repository</i> object representing the Enterprise Architect model about to be closed.</p> <p>Poll its members to retrieve model data and user interface status information.</p> |

#### Return Value:

None.

### 19.3.5.5 EA\_FileNew

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                     | See also                                                                            |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| <b>Details</b> | <p>The <i>EA_FileNew</i> event enables the Add-In to respond to a <i>File New</i> event. When Enterprise Architect creates a new model file, this event is raised and passed to all Add-Ins implementing this method.</p> <p>The event occurs when the model being viewed by the Enterprise Architect user changes, for whatever reason (through user interaction or Add-In activity).</p> | <a href="#">EA_FileClose</a><br>[2615]<br><br><a href="#">EA_FileOpen</a><br>[2615] |

**Syntax:****Sub EA\_FileNew(Repository As EA.Repository)**

The *EA\_FileNew* function syntax contains the following elements:

| Parameter         | Type                                                   | Direction | Description                                                                                                                                                                      |
|-------------------|--------------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[2448]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |

**Return Value:**

None.

**19.3.5.6 EA\_FileOpen****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                     | See also                                                                                                         |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p>The <i>EA_FileOpen</i> event enables the Add-In to respond to a <i>File Open</i> event. When Enterprise Architect opens a new model file, this event is raised and passed to all Add-Ins implementing this method.</p> <p>The event occurs when the model being viewed by the Enterprise Architect user changes, for whatever reason (through user interaction or Add-In activity).</p> | <a href="#">EA_FileClose</a><br><small>[2615]</small><br><br><a href="#">EA_FileNew</a><br><small>[2615]</small> |

**Syntax:****Sub EA\_FileOpen(Repository As EA.Repository)**

The *EA\_FileOpen* function syntax contains the following elements:

| Parameter         | Type                                                   | Direction | Description                                                                                                                                                                      |
|-------------------|--------------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[2448]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |

**Return Value:**

None.

### 19.3.5.7 *EA\_OnPostCloseDiagram*

#### Topics

| Topic          | Detail                                                                        | See also                                               |
|----------------|-------------------------------------------------------------------------------|--------------------------------------------------------|
| <b>Details</b> | <i>EA_OnPostCloseDiagram</i> notifies Add-Ins that a diagram has been closed. | <a href="#">EA_OnPostOpenDiagram</a> <sup>[2618]</sup> |

#### Syntax:

**Function** *EA\_OnPostCloseDiagram(Repository As EA.Repository, DiagramID As Integer)*

The *EA\_OnPostCloseDiagram* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                          |
|-------------------|-------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DiagramID</b>  | <i>Integer</i>                                  | IN        | Contains the Diagram ID of the diagram that was closed.                                                                                                                              |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An <i>EA.Repository</i> object representing the Enterprise Architect model about to be closed.<br><br>Poll its members to retrieve model data and user interface status information. |

#### Return Value:

None.

### 19.3.5.8 *EA\_OnPostInitialized*

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                    | See also                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| <b>Details</b> | <i>EA_OnPostInitialized</i> notifies Add-Ins that the Repository object has finished loading and any necessary initialization steps can now be performed on the object.<br><br>For example, the Add-In can create an Output tab using <i>Repository.CreateOutputTab</i> . | <a href="#">Repository.CreateOutputTab</a> <sup>[2455]</sup> |

#### Syntax:

**Sub** *EA\_OnPostInitialized(Repository As EA.Repository)*

The *EA\_OnPostInitialized* function syntax contains the following elements:

| Parameter         | Type                                 | Direction | Description                                                                                                                                                                      |
|-------------------|--------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b> | <a href="#">EA.Repository</a> [2448] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |

**Return Value:**

None.

### 19.3.5.9 *EA\_OnPostOpenDiagram*

**Topics**

| Topic          | Detail                                                                       | See also                                     |
|----------------|------------------------------------------------------------------------------|----------------------------------------------|
| <b>Details</b> | <i>EA_OnPostOpenDiagram</i> notifies Add-Ins that a diagram has been opened. | <a href="#">EA_OnPostCloseDiagram</a> [2617] |

**Syntax:**

**Function** *EA\_OnPostOpenDiagram(Repository As EA.Repository, DiagramID As Integer)*

The *EA\_OnPostOpenDiagram* function syntax contains the following elements:

| Parameter         | Type                                 | Direction | Description                                                                                                                                                                      |
|-------------------|--------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DiagramID</b>  | <i>Integer</i>                       | IN        | Contains the Diagram ID of the diagram that was opened.                                                                                                                          |
| <b>Repository</b> | <a href="#">EA.Repository</a> [2448] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |

**Return Value:**

None.

### 19.3.5.10 EA\_OnPostTransform

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                            | See also |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <p><i>EA_OnPostTransform</i> notifies Add-Ins that an MDG transformation has taken place with the output in the specified target package.</p> <p>This event occurs when a user runs an MDG transform on one or more target packages; the notification is provided for each transform/target package immediately after all transform processes have completed.</p> |          |

#### Syntax:

**Function** *EA\_OnPostTransform(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPostTransform* function syntax contains the following elements:

| Parameter         | Type                                                        | Direction | Description                                                                                                                                                                                                                                                                                                                             |
|-------------------|-------------------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br><small>[2431]</small> | IN        | <p>Contains the following <i>EventProperty Objects</i> for the transform performed:</p> <ul style="list-style-type: none"> <li><i>Transform</i>: A string value corresponding to the name of the transform used</li> <li><i>PackageID</i>: A long value corresponding to <i>Package.PackageID</i> of the destination package</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[2448]</small>      | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>                                                                                                                                                 |

#### Return Value:

Reserved for future use.

### 19.3.5.11 EA\_OnPreExitInstance

#### Details:

*EA\_OnPreExitInstance* is not currently used.



**Syntax:****Sub EA\_OnPreExitInstance(Repository As EA.Repository)**

The *EA\_OnPreExitInstance* function syntax contains the following element:

| Parameter         | Type                                                   | Direction | Description                                                                                                                                                                      |
|-------------------|--------------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[2448]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |

**Return Value:**

None.

**19.3.5.12 EA\_OnRetrieveModelTemplate****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                            | See also                                                             |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| <b>Details</b> | <i>EA_OnRetrieveModelTemplate</i> requests that an Add-In pass a model template to Enterprise Architect.<br><br>This event occurs when a user executes the <b>Add a New Model Using Wizard</b> command to add a model that has been defined by an MDG Technology. | <a href="#">Incorporate Model Templates</a><br><small>[1355]</small> |

**Syntax:****Function EA\_OnRetrieveModelTemplate(Repository As EA.Repository,sLocation As String) As String**

The *EA\_OnRetrieveModelTemplate* function syntax contains the following elements:

| Parameter         | Type                                                   | Direction | Description                                                                                                                                                                      |
|-------------------|--------------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[2448]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |
| <b>sLocation</b>  | <i>String</i>                                          | IN        | The name of the template requested; this should match the <i>location</i> attribute in the <i>&lt;ModelTemplates&gt;</i> section of an MDG Technology File.                      |

**Return Value:**

Return a string containing the XML export of the model that is being used as a template.

**Example**

```
Public Function EA_OnRetrieveModelTemplate(ByRef Rep As EA.Repository,
ByRef sLocation As String) As String
 Dim sTemplate As String
 Select Case sLocation
 Case "Templates\Template1.xml"
 sTemplate = My.Resources.Template1
 Case "Templates\Template2.xml"
 sTemplate = My.Resources.Template2
 Case "Templates\Template3.xml"
 sTemplate = My.Resources.Template3
 Case Else
 MsgBox("Path for " & sLocation & " not found")
 sTemplate = ""
 End Select
 EA_OnRetrieveModelTemplate = sTemplate
End Function
```

**19.3.5.13 Model Validation Broadcasts****Perform Model Validation from an Add-In:**

Using Enterprise Architect broadcasts, it is possible to define a set of rules that are evaluated when the user instructs Enterprise Architect to perform model validation. An Add-In that performs model validation would involve the following broadcast events:

| Topic                           | Detail                                                                                                                               | Link                                                              |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| <b>EA_OnInitializeUserRules</b> | <i>EA_OnInitializeUserRules</i> is intercepted in order to define rule categories and rules.                                         | <a href="#">EA_OnInitializeUserRules</a><br><small>[2622]</small> |
| <b>EA_OnStartValidation</b>     | <i>EA_OnStartValidation</i> can be intercepted to perform any required processing prior to validation.                               | <a href="#">EA_OnStartValidation</a><br><small>[2623]</small>     |
| <b>Validate Request</b>         | The following functions intercept each request to validate an individual element, package, diagram, connector, attribute and method. |                                                                   |
|                                 | <i>EA_OnRunElementRule</i>                                                                                                           | <a href="#">EA_OnRunElementRule</a><br><small>[2624]</small>      |
|                                 | <i>EA_OnRunPackageRule</i>                                                                                                           | <a href="#">EA_OnRunPackageRule</a><br><small>[2625]</small>      |
|                                 | <i>EA_OnRunDiagramRule</i>                                                                                                           | <a href="#">EA_OnRunDiagramRule</a><br><small>[2625]</small>      |

| Topic                     | Detail                                                                                                        | Link                                                    |
|---------------------------|---------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
|                           | <i>EA_OnRunConnectorRule</i>                                                                                  | <a href="#">EA_OnRunConnectorRule</a> <sup>[2626]</sup> |
|                           | <i>EA_OnRunAttributeRule</i>                                                                                  | <a href="#">EA_OnRunAttributeRule</a> <sup>[2627]</sup> |
|                           | <i>EA_OnRunMethodRule</i>                                                                                     | <a href="#">EA_OnRunMethodRule</a> <sup>[2628]</sup>    |
| <b>EA_OnEndValidation</b> | <i>EA_OnEndValidation</i> can be intercepted to perform any required clean-up after validation has completed. | <a href="#">EA_OnEndValidation</a> <sup>[2623]</sup>    |

#### Learn more

- [Model Validation Example](#) <sup>[2630]</sup>

### 19.3.5.13.1 *EA\_OnInitializeUserRules*

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See also                                                  |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnInitializeUserRules</i> is called on Enterprise Architect start-up and requests that the Add-In provide Enterprise Architect with a rule category and list of rule IDs for model validation.</p> <p>This function must be implemented by any Add-In that is to perform its own model validation. It must call <i>Project.DefineRuleCategory</i> once and <i>Project.DefineRule</i> for each rule; these functions are described in the <i>Project Interface</i> topic.</p> | <a href="#">Project Interface Class</a> <sup>[2535]</sup> |

#### Syntax:

#### **Sub *EA\_OnInitializeUserRules(Repository As EA.Repository)***

The *EA\_OnInitializeUserRules* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                             |
|-------------------|-------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p> |

### 19.3.5.13.2 EA\_OnStartValidation

#### Topics

| Topic          | Detail                                                                                                                       | See also |
|----------------|------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <i>EA_OnStartValidation</i> notifies Add-Ins that a user has invoked the model validation command from Enterprise Architect. |          |

#### Syntax:

**Sub EA\_OnStartValidation(Repository As EA.Repository, ParamArray Args() as Variant)**

The *EA\_OnStartValidation* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                  |
|-------------------|-------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Args</b>       | <i>ParamArray of Variant</i>                    | IN        | Contains a list of Rule Categories that are active for the current invocation of model validation.                                                                           |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br>Poll its members to retrieve model data and user interface status information. |

### 19.3.5.13.3 EA\_OnEndValidation

#### Topics

| Topic          | Detail                                                                                                                                                            | See also |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <i>EA_OnEndValidation</i> notifies Add-Ins that model validation has completed.<br>Use this event to arrange any clean-up operations arising from the validation. |          |

#### Syntax:

**Sub EA\_OnEndValidation(Repository As EA.Repository, ParamArray Args() as Variant)**

The *EA\_OnEndValidation* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                      |
|-------------------|-------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Args</b>       | <i>ParamArray of Variant</i>                    | IN        | Contains a list of Rule Categories that were active for the invocation of model validation that has just completed.                                                              |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |

#### 19.3.5.13.4 EA\_OnRunElementRule

##### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | See also                                                  |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| <b>Details</b> | <p>This event is triggered once for each rule defined in <i>EA_OnInitializeUserRules</i> to be performed on each element in the selection being validated.</p> <p>If you don't want to perform the rule defined by <b>RuleID</b> on the given element, then simply return without performing any action.</p> <p>On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect.</p> | <a href="#">EA_OnInitializeUserRule</a> <sup>[2622]</sup> |

##### Syntax:

**Sub EA\_OnRunElementRule(Repository As EA.Repository, RuleID As String, Element As EA.Element)**

The *EA\_OnRunElementRule* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                      |
|-------------------|-------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Element</b>    | <i>EA.Element</i>                               | IN        | The element to potentially perform validation on.                                                                                                                                |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |
| <b>RuleID</b>     | <i>String</i>                                   | IN        | The ID that was passed into the <i>Project.DefineRule</i> command.                                                                                                               |

### 19.3.5.13.5 EA\_OnRunPackageRule

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | See also                                                            |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| <b>Details</b> | <p>This event is triggered once for each rule defined in <i>EA_OnInitializeUserRules</i> to be performed on each package in the selection being validated.</p> <p>If you don't want to perform the rule defined by <b>RuleID</b> on the given package, then simply return without performing any action.</p> <p>On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect.</p> | <a href="#">EA_OnInitializeUserRule</a><br><a href="#">S [2622]</a> |

#### Syntax:

**Sub EA\_OnRunPackageRule(Repository As EA.Repository, RuleID As String, PackageID As Long)**

The *EA\_OnRunElementRule* function syntax contains the following elements:

| Parameter         | Type                                                    | Direction | Description                                                                                                                                                                  |
|-------------------|---------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PackageID</b>  | <i>Long</i>                                             | IN        | The ID of the package to potentially perform validation on.<br>Use the <i>Repository.GetPackageByID</i> method to retrieve the package object.                               |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><a href="#">[2448]</a> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br>Poll its members to retrieve model data and user interface status information. |
| <b>RuleID</b>     | <i>String</i>                                           | IN        | The ID that was passed into the <i>Project.DefineRule</i> method.                                                                                                            |

### 19.3.5.13.6 EA\_OnRunDiagramRule

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                  | See also                                                            |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| <b>Details</b> | <p>This event is triggered once for each rule defined in <i>EA_OnInitializeUserRules</i> to be performed on each diagram in the selection being validated.</p> <p>If you don't want to perform the rule defined by <b>RuleID</b> on</p> | <a href="#">EA_OnInitializeUserRule</a><br><a href="#">S [2622]</a> |

| Topic | Detail                                                                                                                                                                                                                                           | See also |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>the given diagram, then simply return without performing any action.</p> <p>On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect.</p> |          |

**Syntax:*****Sub EA\_OnRunDiagramRule(Repository As EA.Repository, RuleID As String, DiagramID As Long)***

The *EA\_OnRunDiagramRule* function syntax contains the following elements:

| Parameter         | Type                                                   | Direction | Description                                                                                                                                                                  |
|-------------------|--------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DiagramID</b>  | <i>Long</i>                                            | IN        | The ID of the diagram to potentially perform validation on.<br>Use the <i>Repository.GetDiagramByID</i> method to retrieve the diagram object.                               |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[2448]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br>Poll its members to retrieve model data and user interface status information. |
| <b>RuleID</b>     | <i>String</i>                                          | IN        | The ID that was passed into the <i>Project.DefineRule</i> command.                                                                                                           |

**19.3.5.13.7 EA\_OnRunConnectorRule****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See also                                                          |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| <b>Details</b> | <p>This event is triggered once for each rule defined in <i>EA_OnInitializeUserRules</i> to be performed on each connector in the selection being validated.</p> <p>If you don't want to perform the rule defined by <b>RuleID</b> on the given connector, then simply return without performing any action.</p> <p>On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect.</p> | <a href="#">EA_OnInitializeUserRules</a><br><small>[2622]</small> |

**Syntax:**

**Sub EA\_OnRunConnectorRule(Repository As EA.Repository, RuleID As String, ConnectorID As Long)**

The *EA\_OnRunConnectorRule* function syntax contains the following elements:

| Parameter          | Type                                    | Direction | Description                                                                                                                                                               |
|--------------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ConnectorID</b> | <i>Long</i>                             | IN        | The ID of the connector to potentially perform validation on.<br>Use the <i>Repository.GetConnectorByID</i> method to retrieve the connector object.                      |
| <b>Repository</b>  | <a href="#">EA.Repository</a><br>[2448] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| <b>RuleID</b>      | <i>String</i>                           | IN        | The ID that was passed into the <i>Project.DefineRule</i> command.                                                                                                        |

### 19.3.5.13.8 EA\_OnRunAttributeRule

**Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See also                                            |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| <b>Details</b> | <p>This event is triggered once for each rule defined in <i>EA_OnInitializeUserRules</i> to be performed on each attribute in the selection being validated.</p> <p>If you don't want to perform the rule defined by <b>RuleID</b> on the given attribute, then simply return without performing any action.</p> <p>On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect.</p> | <a href="#">EA_OnInitializeUserRule</a><br>s [2622] |

**Syntax:**

**Sub EA\_OnRunAttributeRule(Repository As EA.Repository, RuleID As String, AttributeGUID As String, ObjectID As Long)**

The *EA\_OnRunAttributeRule* function syntax contains the following elements:

| Parameter            | Type          | Direction | Description                                                                                                                        |
|----------------------|---------------|-----------|------------------------------------------------------------------------------------------------------------------------------------|
| <b>AttributeGUID</b> | <i>String</i> | IN        | The GUID of the attribute to potentially perform validation on.<br>Use the <i>Repository.GetAttributeByGuid</i> method to retrieve |



| Parameter         | Type                                    | Direction | Description                                                                                                                                                                  |
|-------------------|-----------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   |                                         |           | the attribute object.                                                                                                                                                        |
| <b>ObjectID</b>   | <i>Long</i>                             | IN        | The ID of the object that owns the given attribute.<br>Use the <i>Repository.GetObjectByID</i> method to retrieve the object.                                                |
| <b>Repository</b> | <a href="#">EA.Repository</a><br>[2448] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br>Poll its members to retrieve model data and user interface status information. |
| <b>RuleID</b>     | <i>String</i>                           | IN        | The ID that was passed into the <i>Project.DefineRule</i> command.                                                                                                           |

### 19.3.5.13.9 EA\_OnRunMethodRule

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See also                                           |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| <b>Details</b> | <p>This event is triggered once for each rule defined in <i>EA_OnInitializeUserRules</i> to be performed on each method in the selection being validated.</p> <p>If you don't want to perform the rule defined by <b>RuleID</b> on the given method, then simply return without performing any action.</p> <p>On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect.</p> | <a href="#">EA_OnInitializeUserRules</a><br>[2622] |

#### Syntax:

**Sub EA\_OnRunMethodRule(Repository As EA.Repository, RuleID As String, MethodGUID As String, ObjectID As Long)**

The *EA\_OnRunMethodRule* function syntax contains the following elements:

| Parameter         | Type          | Direction | Description                                                                                                                                     |
|-------------------|---------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MethodGUID</b> | <i>String</i> | IN        | The GUID of the method to potentially perform validation on.<br>Use the <i>Repository.GetMethodByGuid</i> method to retrieve the method object. |

| Parameter         | Type                                    | Direction | Description                                                                                                                                                                  |
|-------------------|-----------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ObjectID</b>   | <i>Long</i>                             | IN        | The ID of the object that owns the given method.<br>Use the <i>Repository.GetObjectByID</i> method to retrieve the object.                                                   |
| <b>Repository</b> | <a href="#">EA.Repository</a><br>[2448] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br>Poll its members to retrieve model data and user interface status information. |
| <b>RuleID</b>     | <i>String</i>                           | IN        | The ID that was passed into the <i>Project.DefineRule</i> command.                                                                                                           |

### 19.3.5.13.10 EA\_OnRunParameterRule

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See also                                           |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| <b>Details</b> | <p>This event is triggered once for each rule defined in <i>EA_OnInitializeUserRules</i> to be performed on each parameter in the selection being validated.</p> <p>If you don't want to perform the rule defined by <b>RuleID</b> on the given parameter, then simply return without performing any action.</p> <p>On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect.</p> | <a href="#">EA_OnInitializeUserRules</a><br>[2622] |

#### Syntax:

**Sub EA\_OnRunParameterRule(Repository As EA.Repository, RuleID As String, ParameterGUID As String, MethodGUID As String, ObjectID As Long)**

The *EA\_OnRunMethodRule* function syntax contains the following elements:

| Parameter         | Type          | Direction | Description                                                                                                                              |
|-------------------|---------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MethodGUID</b> | <i>String</i> | IN        | The GUID of the method that owns the given parameter.<br>Use the <i>Repository.GetMethodByGuid</i> method to retrieve the method object. |
| <b>ObjectID</b>   | <i>Long</i>   | IN        | The ID of the object that owns the given parameter.<br>Use the <i>Repository.GetObjectByID</i> method to retrieve the object.            |

| Parameter            | Type                                    | Direction | Description                                                                                                                                                                  |
|----------------------|-----------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ParameterGUID</b> | <i>String</i>                           | IN        | The GUID of the parameter to potentially perform validation on.<br>Use this to retrieve the parameter by iterating through the <i>Method.Parameters</i> collection.          |
| <b>Repository</b>    | <a href="#">EA.Repository</a><br>[2448] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br>Poll its members to retrieve model data and user interface status information. |
| <b>RuleID</b>        | <i>String</i>                           | IN        | The ID that was passed into the <i>Project.DefineRule</i> command.                                                                                                           |

#### 19.3.5.13.11 Model Validation Example

The following example code is written in C# and provides a skeleton model validation implementation that you might like to use as a starting point in writing your own model validation rules.

##### Main.cs:

```
using System;

namespace myAddIn
{
 public class Main
 {
 public Rules theRules;

 public Main()
 {
 theRules = new Rules();
 }

 public string EA_Connect(EA.Repository Repository)
 {
 return "";
 }

 public void EA_Disconnect()
 {
 GC.Collect();
 GC.WaitForPendingFinalizers();
 }

 private bool IsProjectOpen(EA.Repository Repository)
 {
 try
 {
 EA.Collection c = Repository.Models;
 return true;
 }
 catch
 {
 return false;
 }
 }
 }
}
```

```

 }
}

public object EA_GetMenuItems(EA.Repository Repository, string
MenuLocation, string MenuName)
{
 switch (MenuName)
 {
 case "":
 return "- &myAddin";
 case "- &myAddin":
 string() ar = { "&Test" };
 return ar;
 }
 return "";
}

public void EA_GetMenuState(EA.Repository Repository, string
MenuLocation, string MenuName, string ItemName, ref bool IsEnabled, ref
bool IsChecked)
{
 // if no open project, disable all menu options
 if (IsProjectOpen(Repository))
 IsEnabled = true;
 else
 IsEnabled = false;
}

public void EA_MenuClick(EA.Repository Repository, string
MenuLocation, string MenuName, string ItemName)
{
 switch (ItemName)
 {
 case "&Test";
 DoTest(Repository);
 break;
 }
}

public void EA_OnInitializeUserRules(EA.Repository Repository)
{
 if (Repository != null)
 {
 theRules.ConfigureCategories(Repository);
 theRules.ConfigureRules(Repository);
 }
}

public void EA_OnRunElementRule(EA.Repository Repository,
string RuleID, EA.Element element)
{
 theRules.RunElementRule(Repository, RuleID, element);
}

public void EA_OnRunDiagramRule(EA.Repository Repository,
string RuleID, long IDiagramID)
{
 theRules.RunDiagramRule(Repository, RuleID, IDiagramID);
}

public void EA_OnRunConnectorRule(EA.Repository Repository,
string RuleID, long IConnectorID)
{
 theRules.RunConnectorRule(Repository, RuleID,
IConnectorID);
}

public void EA_OnRunAttributeRule(EA.Repository Repository,
string RuleID, string AttGUID, long IObjectID)
{

```

```

 return;
 }

 public void EA_OnDeleteTechnology(EA.Repository Repository, EA.
Event Properties Info)
 {
 return;
 }

 public void EA_OnImportTechnology(EA.Repository Repository, EA.
Event Properties Info)
 {
 return;
 }

 private void DoTest(EA.Repository Rep)
 {
 // TODO: insert test code here
 }
}
}

```

**Rules.cs:**

```

using System;
using System.Collections;

namespace myAddin
{
 public class Rules
 {
 private string m_sCategoryID;
 private System.Collections.ArrayList m_RuleIDs;
 private System.Collections.ArrayList m_RuleDEx;

 private const string cRule01 = "Rule01";
 private const string cRule02 = "Rule02";
 private const string cRule03 = "Rule03";
 // TODO: expand this list as much as necessary

 public Rules()
 {
 m_RuleIDs = new System.Collections.ArrayList();
 m_RuleDEx = new System.Collections.ArrayList();
 }

 private string LookupMap(string sKey)
 {
 return DoLookupMap(sKey, m_RuleIDs, m_RuleDEx);
 }

 private string LookupMapEx(string sRule)
 {
 return DoLookupMap(sRule, m_RuleDEx, m_RuleIDs);
 }

 private string DoLookupMap(string sKey, ArrayList arrValues,
ArrayList arrKeys)
 {
 if (arrKeys.Contains(sKey))
 return arrValues(arrKeys.IndexOf(sKey)) .ToString
();
 else
 return "";
 }

 private void AddToMap(string sRuleID, string sKey)
 {

```

```

 m_RuleIDs.Add(sRuleID);
 m_RuleDEx.Add(sKey);
 }

 private string GetRuleStr(string sRuleID)
 {
 switch (sRuleID)
 {
 case cRule01:
 return "Error Message 01";
 case cRule02:
 return "Error Message 02";
 case cRule03:
 return "Error Message 03";
 // TODO: add extra cases as much as necessary
 }
 return "";
 }

 public void ConfigureCategories(EA.Repository Repository)
 {
 EA.Project Project = Repository.GetProjectInterface();
 m_sCategoryID = Project.DefineRuleCategory("Enterprise
Collaboration Architecture (ECA) Rules");
 }

 public void ConfigureRules(EA.Repository Repository)
 {
 EA.Project Project = Repository.GetProjectInterface();
 AddToMap(Project.DefineRule(m_sCategoryID, EA.
EnumMVErrType.mvError, GetRuleStr(cRule01)), cRule01);
 AddToMap(Project.DefineRule(m_sCategoryID, EA.
EnumMVErrType.mvError, GetRuleStr(cRule02)), cRule02);
 AddToMap(Project.DefineRule(m_sCategoryID, EA.
EnumMVErrType.mvError, GetRuleStr(cRule03)), cRule03);
 // TODO: expand this list
 }

 public void RunConnectorRule(EA.Repository Repository, string
sRuleID, long lConnectorID)
 {
 EA.Connector Connector = Repository.GetConnectorByID
((int)lConnectorID);
 if (Connector != null)
 {
 switch (LookupMapEx(sRuleID))
 {
 case cRule02:
 // TODO: perform rule 2 check
 break;
 // TODO: add more cases
 }
 }
 }

 public void RunDiagramRule(EA.Repository Repository, string
sRuleID, long lDiagramID)
 {
 EA.Diagram Diagram = Repository.GetDiagramByID((int)
lDiagramID);
 if (Diagram != null)
 {
 switch (LookupMapEx(sRuleID))
 {
 case cRule03:
 // TODO: perform rule 3 check
 break;
 // TODO: add more cases
 }
 }
 }
}

```

```

public void RunElementRule(EA.Repository Repository, string
sRuleID, EA.Element Element)
{
 if (Element != null)
 {
 switch (LookupMapEx(sRuleID))
 {
 case cRule01:
 DoRule01(Repository, Element);
 break;
 // TODO: add more cases
 }
 }
}

private void DoRule01(EA.Repository Repository, EA.Element
Element)
{
 if (Element.Stereotype != "myStereotype")
 return;

 // TODO: validation logic here

 // report validation errors
 EA.Project Project = Repository.GetProjectInterface();
 Project.PublishResult(LookupMap(cRule01), EA.
EnumMErrorType.mvError, GetRuleStr(cRule01));
}
}
}

```

#### 19.3.5.14 Post-New Events

Enterprise Architect Add-Ins can respond to the creation of new elements, connectors, objects, attributes, methods and packages using the following broadcast events:

| Topic                     | Link                                                        |
|---------------------------|-------------------------------------------------------------|
| EA_OnPostNewElement       | <a href="#">EA_OnPostNewElement</a> <sup>[2635]</sup>       |
| EA_OnPostNewConnector     | <a href="#">EA_OnPostNewConnector</a> <sup>[2636]</sup>     |
| EA_OnPostNewDiagram       | <a href="#">EA_OnPostNewDiagram</a> <sup>[2637]</sup>       |
| EA_OnPostNewDiagramObject | <a href="#">EA_OnPostNewDiagramObject</a> <sup>[2638]</sup> |
| EA_OnPostNewAttribute     | <a href="#">EA_OnPostNewAttribute</a> <sup>[2639]</sup>     |
| EA_OnPostNewMethod        | <a href="#">EA_OnPostNewMethod</a> <sup>[2640]</sup>        |
| EA_OnPostNewPackage       | <a href="#">EA_OnPostNewPackage</a> <sup>[2641]</sup>       |
| EA_OnPostNewGlossaryTerm  | <a href="#">EA_OnPostNewGlossaryTerm</a> <sup>[2642]</sup>  |

| Topic | Link |
|-------|------|
|       |      |

### 19.3.5.14.1 EA\_OnPostNewElement

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See also                                             |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPostNewElement</i> notifies Add-Ins that a new element has been created on a diagram. It enables Add-Ins to modify the element upon creation.</p> <p>This event occurs after a user has dragged a new element from the Toolbox or Resources window onto a diagram. The notification is provided immediately after the element is added to the model.</p> <p>Set <i>Repository.SuppressEADialogs</i> to <b>true</b> to suppress Enterprise Architect from showing its default dialogs.</p> | <a href="#">EA_OnPreNewElement</a> <sup>[2651]</sup> |

#### Syntax:

**Function** *EA\_OnPostNewElement*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewElement* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                                |
|-------------------|------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[2431]</sup> | IN        | <p>Contains the following <i>EventProperty</i> objects for the new element:</p> <ul style="list-style-type: none"> <li><i>ElementID</i>: A long value corresponding to <i>Element.ElementID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup>      | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>                    |

#### Return Value:

Return **True** if the element has been updated during this notification. Return **False** otherwise.



### 19.3.5.14.2 EA\_OnPostNewConnector

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also                                               |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPostNewConnector</i> notifies Add-Ins that a new connector has been created on a diagram. It enables Add-Ins to modify the connector upon creation.</p> <p>This event occurs after a user has dragged a new connector from the Toolbox or Resources window onto a diagram. The notification is provided immediately after the connector is added to the model.</p> <p>Set <i>Repository.SuppressEADialogs</i> to <b>true</b> to suppress Enterprise Architect from showing its default dialogs.</p> | <a href="#">EA_OnPreNewConnector</a> <sup>[2652]</sup> |

#### Syntax:

**Function** *EA\_OnPostNewConnector* (*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewConnector* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                                 |
|-------------------|------------------------------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[2437]</sup> | IN        | Contains the following <i>EventProperty</i> objects for the new connector: <ul style="list-style-type: none"> <li><i>ConnectorID</i>: A long value corresponding to <i>Connector.ConnectorID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information.                            |

#### Return Value:

Return **True** if the connector has been updated during this notification. Return **False** otherwise.

### 19.3.5.14.3 EA\_OnPostNewDiagram

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                    | See also                                             |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPostNewDiagram</i> notifies Add-Ins that a new diagram has been created. It enables Add-Ins to modify the diagram upon creation.</p> <p>Set <i>Repository.SuppressEADialogs</i> to true to suppress Enterprise Architect from showing its default dialogs.</p> | <a href="#">EA_OnPreNewDiagram</a> <sup>[2653]</sup> |

#### Syntax:

**Function** *EA\_OnPostNewDiagram*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewDiagram* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                                                 |
|-------------------|------------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[2437]</sup> | IN        | <p>Contains the following <i>EventProperty</i> objects for the new diagram:</p> <ul style="list-style-type: none"> <li><i>DiagramID</i>: A long value corresponding to <i>Diagram</i>.</li> <li><i>PackageID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup>      | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>                                     |

#### Return Value:

Return **True** if the diagram has been updated during this notification. Return **False** otherwise.

#### 19.3.5.14.4 EA\_OnPostNewDiagramObject

##### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See also                                                   |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPostNewDiagramObject</i> notifies Add-Ins that a new object has been created on a diagram. It enables Add-Ins to modify the object upon creation.</p> <p>This event occurs after a user has dragged a new object from the Project Browser or Resources window onto a diagram. The notification is provided immediately after the object is added to the diagram.</p> <p>Set <i>Repository.SuppressEADialogs</i> to <b>true</b> to suppress Enterprise Architect from showing its default dialogs.</p> | <a href="#">EA_OnPreNewDiagramObject</a> <sup>[2654]</sup> |

##### Syntax:

**Function** EA\_OnPostNewDiagramObject(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewDiagramObject* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                      |
|-------------------|------------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[2431]</sup> | IN        | Contains the following <i>EventProperty</i> objects for the new element: <ul style="list-style-type: none"> <li><i>ObjectID</i>: A long value corresponding to <i>Object.ObjectID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information.                 |

##### Return Value:

Return **True** if the element has been updated during this notification. Return **False** otherwise.

### 19.3.5.14.5 EA\_OnPostNewAttribute

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also                                       |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPostNewAttribute</i> notifies Add-Ins that a new attribute has been created on a diagram. It enables Add-Ins to modify the attribute upon creation.</p> <p>This event occurs when a user creates a new attribute on an element by either drag-dropping from the Project Browser, using the Attributes Properties dialog, or using the in-place editor on the diagram. The notification is provided immediately after the attribute is created.</p> <p>Set <i>Repository.SuppressEADialogs</i> to <b>true</b> to suppress Enterprise Architect from showing its default dialogs.</p> | <a href="#">EA_OnPreNewAttribute</a><br>[2655] |

#### Syntax:

**Function** *EA\_OnPostNewAttribute*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewAttribute* function syntax contains the following elements:

| Parameter         | Type                                         | Direction | Description                                                                                                                                                                                                |
|-------------------|----------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br>[2431] | IN        | Contains the following <i>EventProperty</i> objects for the new attribute: <ul style="list-style-type: none"> <li><i>AttributeID</i>: A long value corresponding to <i>AttributeAttributeID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br>[2448]      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information.                           |

#### Return Value:

Return **True** if the attribute has been updated during this notification. Return **False** otherwise.

### 19.3.5.14.6 EA\_OnPostNewMethod

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See also                                                   |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPostNewMethod</i> notifies Add-Ins that a new method has been created on a diagram. It enables Add-Ins to modify the method upon creation.</p> <p>This event occurs when a user creates a new method on an element by either drag-dropping from the Project Browser, using the method's Properties dialog, or using the in-place editor on the diagram. The notification is provided immediately after the method is created.</p> <p>Set <i>Repository.SuppressEADialogs</i> to true to suppress Enterprise Architect from showing its default dialogs.</p> | <a href="#">EA_OnPreNewMethod</a><br><small>[2656]</small> |

#### Syntax:

**Function** *EA\_OnPostNewMethod*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewMethod* function syntax contains the following elements:

| Parameter         | Type                                                        | Direction | Description                                                                                                                                                                                     |
|-------------------|-------------------------------------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br><small>[2431]</small> | IN        | Contains the following <i>EventProperty</i> objects for the new method: <ul style="list-style-type: none"> <li><i>MethodID</i>: A long value corresponding to <i>Method.MethodID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[2448]</small>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information.                |

#### Return Value:

Return **True** if the method has been updated during this notification. Return **False** otherwise.

### 19.3.5.14.7 EA\_OnPostNewPackage

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | See also                                     |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPostNewPackage</i> notifies Add-Ins that a new package has been created on a diagram. It enables Add-Ins to modify the package upon creation.</p> <p>This event occurs when a user drags a new package from the Toolbox or Resources window onto a diagram, or by selecting the <b>New Package</b> icon from the Project Browser.</p> <p>Set <i>Repository.SuppressEADialogs</i> to <b>true</b> to suppress Enterprise Architect from showing its default dialogs.</p> | <a href="#">EA_OnPreNewPackage</a><br>[2657] |

#### Syntax:

**Function** *EA\_OnPostNewPackage*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewPackage* function syntax contains the following elements:

| Parameter         | Type                                         | Direction | Description                                                                                                                                                                                                     |
|-------------------|----------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br>[2437] | IN        | Contains the following <i>EventProperty</i> objects for the new package: <ul style="list-style-type: none"> <li><i>PackageID</i>: A long value corresponding to <i>Package</i>.<br/><i>PackageID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br>[2448]      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information.                                |

#### Return Value:

Return **True** if the package has been updated during this notification. Return **False** otherwise.

### 19.3.5.14.8 EA\_OnPostNewGlossaryTerm

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                           | See also                                                  |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPostNewGlossaryTerm</i> notifies Add-Ins that a new glossary term has been created. It enables Add-Ins to modify the glossary term upon creation.</p> <p>The notification is provided immediately after the glossary term is added to the model.</p> <p>Set <i>Repository.SuppressEADialogs</i> to <b>true</b> to suppress Enterprise Architect from showing its default dialogs.</p> | <a href="#">EA_OnPreNewGlossaryTerm</a> <sup>[2658]</sup> |

#### Syntax:

**Function** *EA\_OnPostNewGlossaryTerm*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewGlossaryTerm* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                                     |
|-------------------|------------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[2431]</sup> | IN        | <p>Contains the following <i>EventProperty</i> objects for the new glossary term:</p> <ul style="list-style-type: none"> <li><i>ElementID</i>: A long value corresponding to <i>ElementElementID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup>      | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>                         |

#### Return Value:

Return **True** if the glossary term has been updated during this notification. Return **False** otherwise.

### 19.3.5.15 Pre-Deletion Events

Enterprise Architect Add-Ins can respond to requests to delete elements, attributes, methods, connectors, diagrams, packages and glossary terms using the following broadcast events:

| Topic                 | Link                                                    |
|-----------------------|---------------------------------------------------------|
| EA_OnPreDeleteElement | <a href="#">EA_OnPreDeleteElement</a> <sup>[2643]</sup> |

| Topic                                 | Link                                                         |
|---------------------------------------|--------------------------------------------------------------|
| EA_OnPreDeleteAttribute               | <a href="#">EA_OnPreDeleteAttribute</a> <sup>[2644]</sup>    |
| EA_OnPreDeleteMethod                  | <a href="#">EA_OnPreDeleteMethod</a> <sup>[2645]</sup>       |
| EA_OnPreDeleteConnector               | <a href="#">EA_OnPreDeleteConnector</a> <sup>[2646]</sup>    |
| EA_OnPreDeleteDiagram                 | <a href="#">EA_OnPreDeleteDiagram</a> <sup>[2647]</sup>      |
| EA_OnPreDeletePackage                 | <a href="#">EA_OnPreDeletePackage</a> <sup>[2648]</sup>      |
| EA_OnPreDeleteGlossaryTerm            | <a href="#">EA_OnPreDeleteGlossaryTerm</a> <sup>[2649]</sup> |
| EA_OnPreDeleteTechnology (Deprecated) | <a href="#">EA_OnPreDeleteTechnology</a> <sup>[2666]</sup>   |

### 19.3.5.15.1 EA\_OnPreDeleteElement

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                | See also |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <p><i>EA_OnPreDeleteElement</i> notifies Add-Ins that an element is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the element.</p> <p>This event occurs when a user deletes an element from the Project Browser or on a diagram. The notification is provided immediately before the element is deleted, so that the Add-In can disable deletion of the element.</p> |          |

#### Syntax:

**Function** EA\_OnPreDeleteElement(*Repository As EA.Repository, Info As EA.EventProperties*) As **Boolean**

The *EA\_OnPreDeleteElement* function syntax contains the following elements:

| Parameter   | Type                                                 | Direction | Description                                                                                                                                                                                                 |
|-------------|------------------------------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b> | <a href="#">EA.EventProperties</a> <sup>[2431]</sup> | IN        | <p>Contains the following <i>EventProperty</i> objects for the element to be deleted:</p> <ul style="list-style-type: none"> <li><i>ElementID</i>: A long value corresponding to <i>Element</i>.</li> </ul> |



| Parameter         | Type                                            | Direction | Description                                                                                                                                                               |
|-------------------|-------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   |                                                 |           | <i>ElementID</i>                                                                                                                                                          |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An EA.Repository object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |

**Return Value:**

Return **True** to enable deletion of the element from the model. Return **False** to disable deletion of the element.

**19.3.5.15.2 EA\_OnPreDeleteAttribute****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                   | See also |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <p><i>EA_OnPreDeleteAttribute</i> notifies Add-Ins that an attribute is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the attribute.</p> <p>This event occurs when a user attempts to permanently delete an attribute from the Project Browser. The notification is provided immediately before the attribute is deleted, so that the Add-In can disable deletion of the attribute.</p> |          |

**Syntax:**

**Function** *EA\_OnPreDeleteAttribute*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreDeleteAttribute* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                                                       |
|-------------------|------------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[2431]</sup> | IN        | Contains the following <i>EventProperty</i> objects for the attribute to be deleted: <ul style="list-style-type: none"> <li><i>AttributeID</i>: A long value corresponding to <i>Attribute</i>.<br/><i>AttributeID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface                                                                      |

| Parameter | Type | Direction | Description         |
|-----------|------|-----------|---------------------|
|           |      |           | status information. |

**Return Value:**

Return **True** to enable deletion of the attribute from the model. Return **False** to disable deletion of the attribute.

**19.3.5.15.3 EA\_OnPreDeleteMethod****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                           | See also |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <p><b>EA_OnPreDeleteMethod</b> notifies Add-Ins that a method (operation) is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the method.</p> <p>This event occurs when a user attempts to permanently delete a method from the Project Browser. The notification is provided immediately before the method is deleted, so that the Add-In can disable deletion of the method.</p> |          |

**Syntax:**

**Function** EA\_OnPreDeleteMethod(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreDeleteMethod* function syntax contains the following elements:

| Parameter         | Type                                                        | Direction | Description                                                                                                                                                                                                           |
|-------------------|-------------------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br><small>[2431]</small> | IN        | Contains the following <i>EventProperty</i> objects for the method to be deleted: <ul style="list-style-type: none"> <li><i>MethodID</i>: A long value corresponding to <i>Method</i>.<br/><i>MethodID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[2448]</small>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information.                                      |

**Return Value:**

Return **True** to enable deletion of the method from the model. Return **False** to disable deletion of the method.

#### 19.3.5.15.4 EA\_OnPreDeleteConnector

##### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                     | See also |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <p><i>EA_OnPreDeleteConnector</i> notifies Add-Ins that a connector is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the connector.</p> <p>This event occurs when a user attempts to permanently delete a connector on a diagram. The notification is provided immediately before the connector is deleted, so that the Add-In can disable deletion of the connector.</p> |          |

##### Syntax:

**Function** *EA\_OnPreDeleteConnector*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreDeleteConnector* function syntax contains the following elements:

| Parameter         | Type                                                        | Direction | Description                                                                                                                                                                                                           |
|-------------------|-------------------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br><small>[2431]</small> | IN        | Contains the following <i>EventProperty</i> objects for the connector to be deleted: <ul style="list-style-type: none"> <li><i>ConnectorID</i>: A long value corresponding to <i>Connector.ConnectorID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[2448]</small>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information.                                      |

##### Return Value:

Return **True** to enable deletion of the connector from the model. Return **False** to disable deletion of the connector.

### 19.3.5.15.5 EA\_OnPreDeleteDiagram

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                     | See also |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <p><i>EA_OnPreDeleteDiagram</i> notifies Add-Ins that a diagram is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the diagram.</p> <p>This event occurs when a user attempts to permanently delete a diagram from the Project Browser. The notification is provided immediately before the diagram is deleted, so that the Add-In can disable deletion of the diagram.</p> |          |

#### Syntax:

**Function** EA\_OnPreDeleteDiagram(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreDeleteDiagram* function syntax contains the following elements:

| Parameter         | Type                                                       | Direction | Description                                                                                                                                                                                                                      |
|-------------------|------------------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br><small>[243]</small> | IN        | <p>Contains the following <i>EventProperty</i> objects for the diagram to be deleted:</p> <ul style="list-style-type: none"> <li><i>DiagramID</i>: A long value corresponding to <i>Diagram</i>.<br/><i>DiagramID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[244]</small>      | IN        | <p>An <i>EA.Repository</i> object representing the currently-open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>                                          |

#### Return Value:

Return **True** to enable deletion of the diagram from the model. Return **False** to disable deletion of the diagram.

### 19.3.5.15.6 EA\_OnPreDeleteDiagramObject

#### Topics

| Topic          | Detail                                                                                                                                         | See also |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <p><i>EA_OnPreDeleteDiagramObject</i> notifies Add-Ins that a diagram object is to be deleted from the model. It enables Add-Ins to permit</p> |          |

| Topic | Detail                                                                                                                                                                                                                                                                 | See also |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>or deny deletion of the element.</p> <p>This event occurs when a user attempts to permanently delete an element from a diagram. The notification is provided immediately before the element is deleted, so that the Add-In can disable deletion of the element.</p> |          |

**Syntax:**

**Function** *EA\_OnPreDeleteDiagramObject(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreDeleteDiagramObject* function syntax contains the following elements:

| Parameter         | Type                                         | Direction | Description                                                                                                                                                                                                  |
|-------------------|----------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br>[2437] | IN        | <p>Contains the following <i>EventProperty</i> objects for the element to be deleted:</p> <ul style="list-style-type: none"> <li>ID: A long value corresponding to <i>DiagramObject.ElementID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br>[2448]      | IN        | <p>An <i>EA.Repository</i> object representing the currently-open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>                      |

**Return Value:**

Return **True** to enable deletion of the element from the model. Return **False** to disable deletion of the element.

**19.3.5.15.7 EA\_OnPreDeletePackage****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                     | See also |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <p><i>EA_OnPreDeletePackage</i> notifies Add-Ins that a package is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the package.</p> <p>This event occurs when a user attempts to permanently delete a package from the Project Browser. The notification is provided immediately before the package is deleted, so that the Add-In can disable deletion of the package.</p> |          |

**Syntax:**

**Function** *EA\_OnPreDeletePackage(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreDeletePackage* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                                               |
|-------------------|------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[2431]</sup> | IN        | Contains the following <i>EventProperty</i> objects for the package to be deleted: <ul style="list-style-type: none"> <li><i>PackageID</i>: A long value corresponding to <i>Package</i>.<br/><i>PackageID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2445]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information.                                          |

**Return Value:**

Return **True** to enable deletion of the package from the model. Return **False** to disable deletion of the package.

**19.3.5.15.8 EA\_OnPreDeleteGlossaryTerm****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                          | See also |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <i>EA_OnPreDeleteGlossaryTerm</i> notifies Add-Ins that a glossary term is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the glossary term.<br><br>The notification is provided immediately before the glossary term is deleted, so that the Add-In can disable deletion of the glossary term. |          |

**Syntax:**

**Function** *EA\_OnPreDeleteGlossaryTerm(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreDeleteGlossaryTerm* function syntax contains the following elements:

| Parameter         | Type                                         | Direction | Description                                                                                                                                                                                                |
|-------------------|----------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br>[2431] | IN        | Contains the following <i>EventProperty</i> objects for the glossary term to be deleted: <ul style="list-style-type: none"> <li><i>TermID</i>: A long value corresponding to <i>Term.TermID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br>[2448]      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information.                           |

**Return Value:**

Return **True** to enable deletion of the glossary term from the model. Return **False** to disable deletion of the glossary term.

**19.3.5.16 Pre-New Events**

Enterprise Architect Add-Ins can respond to requests to create new elements, connectors, objects, attributes, methods and packages using the following broadcast events:

| Topic                    | Link                                               |
|--------------------------|----------------------------------------------------|
| EA_OnPreNewElement       | <a href="#">EA_OnPreNewElement</a><br>[2651]       |
| EA_OnPreNewConnector     | <a href="#">EA_OnPreNewConnector</a><br>[2652]     |
| EA_OnPreNewDiagram       | <a href="#">EA_OnPreNewDiagram</a><br>[2653]       |
| EA_OnPreNewDiagramObject | <a href="#">EA_OnPreNewDiagramObject</a><br>[2654] |
| EA_OnPreNewAttribute     | <a href="#">EA_OnPreNewAttribute</a><br>[2655]     |
| EA_OnPreNewMethod        | <a href="#">EA_OnPreNewMethod</a><br>[2656]        |
| EA_OnPreNewPackage       | <a href="#">EA_OnPreNewPackage</a><br>[2657]       |
| EA_OnPreNewGlossaryTerm  | <a href="#">EA_OnPreNewGlossaryTerm</a><br>[2658]  |

### 19.3.5.16.1 EA\_OnPreNewElement

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                    | See also                                              |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPreNewElement</i> notifies Add-Ins that a new element is about to be created on a diagram. It enables Add-Ins to permit or deny creation of the new element.</p> <p>This event occurs when a user drags a new element from the Toolbox or Resources window onto a diagram. The notification is provided immediately before the element is created, so that the Add-In can disable addition of the element.</p> | <a href="#">EA_OnPostNewElement</a> <sup>[2635]</sup> |

#### Syntax:

**Function** *EA\_OnPreNewElement*(*Repository* As *EA.Repository*, *Info* As *EA.EventProperties*) As *Boolean*

The *EA\_OnPreNewElement* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------|------------------------------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[2437]</sup> | IN        | <p>Contains the following <i>EventProperty</i> objects for the element to be created:</p> <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Element.Type</i></li> <li><i>Stereotype</i>: A string value corresponding to <i>Element.Stereotype</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Element.ParentID</i></li> <li><i>DiagramID</i>: A long value corresponding to the ID of the diagram to which the element is being added</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup>      | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>                                                                                                                                                                                                                                                                                                               |

#### Return Value:

Return **True** to enable addition of the new element to the model. Return **False** to disable addition of the new element.



### 19.3.5.16.2 EA\_OnPreNewConnector

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                               | See also                                                |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPreNewConnector</i> notifies Add-Ins that a new connector is about to be created on a diagram. It enables Add-Ins to permit or deny creation of a new connector.</p> <p>This event occurs when a user drags a new connector from the Toolbox or Resources window, onto a diagram. The notification is provided immediately before the connector is created, so that the Add-In can disable addition of the connector.</p> | <a href="#">EA_OnPostNewConnector</a> <sup>[2636]</sup> |

#### Syntax:

**Function** *EA\_OnPreNewConnector* (*Repository As EA.Repository, Info As EA.EventProperties*) *As Boolean*

The *EA\_OnPreNewConnector* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------|------------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[2431]</sup> | IN        | <p>Contains the following <i>EventProperty</i> objects for the connector to be created:</p> <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Connector.Type</i></li> <li><i>Subtype</i>: A string value corresponding to <i>Connector.Subtype</i></li> <li><i>Stereotype</i>: A string value corresponding to <i>Connector.Stereotype</i></li> <li><i>ClientID</i>: A long value corresponding to <i>Connector.ClientID</i></li> <li><i>SupplierID</i>: A long value corresponding to <i>Connector.SupplierID</i></li> <li><i>DiagramID</i>: A long value corresponding to <i>Connector.DiagramID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup>      | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

**Return Value:**

Return **True** to enable addition of the new connector to the model. Return **False** to disable addition of the new connector.

**19.3.5.16.3 EA\_OnPreNewDiagram****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                | See also                                              |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPreNewDiagram</i> notifies Add-Ins that a new diagram is about to be created. It enables Add-Ins to permit or deny creation of the new diagram.</p> <p>The notification is provided immediately before the diagram is created, so that the Add-In can disable addition of the diagram.</p> | <a href="#">EA_OnPostNewDiagram</a> <sup>[2637]</sup> |

**Syntax:**

**Function** *EA\_OnPreNewDiagram*(*Repository* As *EA.Repository*, *Info* As *EA.EventProperties*) As *Boolean*

The *EA\_OnPreNewDiagram* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|------------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[2437]</sup> | IN        | <p>Contains the following <i>EventProperty</i> objects for the diagram to be created:</p> <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Diagram.Type</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Diagram.ParentID</i></li> <li><i>PackageID</i>: A long value corresponding to <i>Diagram.PackageID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup>      | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>                                                                                                                                                                                        |

**Return Value:**

Return **True** to enable addition of the new diagram to the model. Return **False** to disable addition of the new diagram.

### 19.3.5.16.4 EA\_OnPreNewDiagramObject

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                       | See also                                                    |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPreNewDiagramObject</i> notifies Add-Ins that a new diagram object is about to be dropped on a diagram. It enables Add-Ins to permit or deny creation of the new object.</p> <p>This event occurs when a user drags an object from the Enterprise Architect Project Browser or Resources window onto a diagram. The notification is provided immediately before the object is created, so that the Add-In can disable addition of the object.</p> | <a href="#">EA_OnPostNewDiagramObject</a> <sup>[2638]</sup> |

#### Syntax:

**Function** EA\_OnPreNewDiagramObject(*Repository As EA.Repository, Info As EA.EventProperties*) As Boolean

The *EA\_OnPreNewDiagramObject* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|------------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[2431]</sup> | IN        | <p>Contains the following <i>EventProperty</i> objects for the object to be created:</p> <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Object.Type</i></li> <li><i>Stereotype</i>: A string value corresponding to <i>Object.Stereotype</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Object.ParentID</i></li> <li><i>DiagramID</i>: A long value corresponding to the ID of the diagram to which the object is being added</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup>      | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>                                                                                                                                                                                                                                                                                                          |

#### Return Value:

Return **True** to enable addition of the object to the model. Return **False** to disable addition of the object.

### 19.3.5.16.5 EA\_OnPreNewAttribute

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See also                                                       |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPreNewAttribute</i> notifies Add-Ins that a new attribute is about to be created on an element. It enables Add-Ins to permit or deny creation of the new attribute.</p> <p>This event occurs when a user creates a new attribute on an element by either drag-dropping from the Project Browser, using the Attributes Properties dialog, or using the in-place editor on the diagram. The notification is provided immediately before the attribute is created, so that the Add-In can disable addition of the attribute.</p> | <a href="#">EA_OnPostNewAttribute</a><br><small>[2639]</small> |

#### Syntax:

**Function** *EA\_OnPreNewAttribute(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreNewAttribute* function syntax contains the following elements:

| Parameter         | Type                                                        | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|-------------------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br><small>[2437]</small> | IN        | <p>Contains the following <i>EventProperty</i> objects for the attribute to be created:</p> <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Attribute.Type</i></li> <li><i>Stereotype</i>: A string value corresponding to <i>Attribute.Stereotype</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Attribute.ParentID</i></li> <li><i>ClassifierID</i>: A long value corresponding to <i>Attribute.ClassifierID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[2448]</small>      | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>                                                                                                                                                                                                                                                                                              |

#### Return Value:

Return **True** to enable addition of the new attribute to the model. Return **False** to disable addition of the new attribute.

### 19.3.5.16.6 EA\_OnPreNewMethod

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See also                                     |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPreNewMethod</i> notifies Add-Ins that a new method is about to be created on an element. It enables Add-Ins to permit or deny creation of the new method.</p> <p>This event occurs when a user creates a new method on an element by either drag-dropping from the Project Browser, using the method Properties dialog, or using the in-place editor on the diagram. The notification is provided immediately before the method is created, so that the Add-In can disable addition of the method.</p> | <a href="#">EA_OnPostNewMethod</a><br>[2640] |

#### Syntax:

**Function** *EA\_OnPreNewMethod(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreNewMethod* function syntax contains the following elements:

| Parameter         | Type                                         | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------|----------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br>[2431] | IN        | <p>Contains the following <i>EventProperty</i> objects for the method to be created:</p> <ul style="list-style-type: none"> <li><i>ReturnType</i>: A string value corresponding to <i>Method.ReturnType</i></li> <li><i>Stereotype</i>: A string value corresponding to <i>Method.Stereotype</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Method.ParentID</i></li> <li><i>ClassifierID</i>: A long value corresponding to <i>Method.ClassifierID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br>[2448]      | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>                                                                                                                                                                                                                                                                                           |

#### Return Value:

Return **True** to enable addition of the new method to the model. Return **False** to disable addition of the new method.

### 19.3.5.16.7 EA\_OnPreNewPackage

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See also                                              |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPreNewPackage</i> notifies Add-Ins that a new package is about to be created in the model. It enables Add-Ins to permit or deny creation of the new package.</p> <p>This event occurs when a user drags a new package from the Toolbox or Resources window onto a diagram, or by selecting the <b>New Package</b> icon from the Project Browser. The notification is provided immediately before the package is created, so that the Add-In can disable addition of the package.</p> | <a href="#">EA_OnPostNewPackage</a> <sup>[2647]</sup> |

#### Syntax:

**Function** *EA\_OnPreNewPackage(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreNewPackage* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------|------------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[2437]</sup> | IN        | <p>Contains the following <i>EventProperty</i> objects for the package to be created:</p> <ul style="list-style-type: none"> <li><i>Stereotype</i>: A string value corresponding to <i>Package.Stereotype</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Package.ParentID</i></li> <li><i>DiagramID</i>: A long value corresponding to the ID of the diagram to which the package is being added</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup>      | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>                                                                                                                                                                                                                                     |

#### Return Value:

Return **True** to enable addition of the new package to the model. Return **False** to disable addition of the new package.

### 19.3.5.16.8 EA\_OnPreNewGlossaryTerm

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                       | See also                                                   |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPreNewGlossaryTerm</i> notifies Add-Ins that a new glossary term is about to be created. It enables Add-Ins to permit or deny creation of the new glossary term.</p> <p>The notification is provided immediately before the glossary term is created, so that the Add-In can disable addition of the element.</p> | <a href="#">EA_OnPostNewGlossaryTerm</a> <sup>[2642]</sup> |

#### Syntax:

**Function** *EA\_OnPreNewGlossaryTerm(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreNewGlossaryTerm* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                               |
|-------------------|------------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[2431]</sup> | IN        | Contains the following <i>EventProperty</i> object for the glossary term to be created: <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Term.TermID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information.                          |

#### Return Value:

Return **True** to enable addition of the new glossary term to the model. Return **False** to disable addition of the new glossary term.

### 19.3.5.17 Tagged Value Broadcasts

Enterprise Architect includes the *Addin Broadcast* Tagged Value type that allows an Add-In to respond to attempts to edit it. The function that is called depends on the type of object the Tagged Value is on.

| Topic                 | Link                                                    |
|-----------------------|---------------------------------------------------------|
| EA_OnAttributeTagEdit | <a href="#">EA_OnAttributeTagEdit</a> <sup>[2659]</sup> |
| EA_OnConnectorTagEdit | <a href="#">EA_OnConnectorTagEdit</a> <sup>[2660]</sup> |

| Topic               | Link                                                  |
|---------------------|-------------------------------------------------------|
| EA_OnElementTagEdit | <a href="#">EA_OnElementTagEdit</a> <sup>[2661]</sup> |
| EA_OnMethodTagEdit  | <a href="#">EA_OnMethodTagEdit</a> <sup>[2662]</sup>  |

### 19.3.5.17.1 EA\_OnAttributeTagEdit

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                | See also                                                                                                                                                                                                                                                                                                               |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnAttributeTagEdit</i> is called when the user clicks the ellipsis ( ... ) button for a Tagged Value of type AddinBroadcast on an attribute.</p> <p>The Add-In displays fields to show and change the value and notes; this function provides the initial values for the Tagged Value notes and value, and takes on any changes on exit of the function.</p> | <p><a href="#">Predefined Structured Types</a><sup>[1386]</sup></p> <p><a href="#">EA_OnAttributeTagEdit</a><sup>[2659]</sup></p> <p><a href="#">EA_OnConnectorTagEdit</a><sup>[2660]</sup></p> <p><a href="#">EA_OnElementTagEdit</a><sup>[2661]</sup></p> <p><a href="#">EA_OnMethodTagEdit</a><sup>[2662]</sup></p> |

#### Syntax:

**Sub EA\_OnAttributeTagEdit(Repository As EA.Repository, AttributeID As Long, String TagName, String TagValue, String TagNotes)**

The *EA\_OnAttributeTagEdit* function syntax contains the following elements:

| Parameter          | Type                                            | Direction | Description                                                                                                                                                                  |
|--------------------|-------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AttributeID</b> | Long                                            | IN        | The ID of the attribute that this Tagged Value is on.                                                                                                                        |
| <b>Repository</b>  | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br>Poll its members to retrieve model data and user interface status information. |
| <b>TagName</b>     | String                                          | IN        | The name of the Tagged Value to edit.                                                                                                                                        |
| <b>TagNotes</b>    | String                                          | INOUT     | The current value of the Tagged Value notes; if the value is                                                                                                                 |



| Parameter       | Type   | Direction | Description                                                                                                               |
|-----------------|--------|-----------|---------------------------------------------------------------------------------------------------------------------------|
|                 |        |           | updated, the new value is stored in the repository on exit of the function.                                               |
| <b>TagValue</b> | String | INOUT     | The current value of the tag; if the value is updated, the new value is stored in the repository on exit of the function. |

### 19.3.5.17.2 EA\_OnConnectorTagEdit

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                      | See also                                                                                                                                                                                                                                                     |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnConnectorTagEdit</i> is called when the user clicks the ellipsis ( ... ) button for a Tagged Value of type <i>AddinBroadcast</i> on a connector.</p> <p>The Add-In displays fields to show and change the value and notes; this function provides the initial values for the Tagged Value notes and value, and takes on any changes on exit of the function.</p> | <p><a href="#">Predefined Structured Types</a> <sup>[1386]</sup></p> <p><a href="#">EA_OnAttributeTagEdit</a> <sup>[2659]</sup></p> <p><a href="#">EA_OnElementTagEdit</a> <sup>[2661]</sup></p> <p><a href="#">EA_OnMethodTagEdit</a> <sup>[2662]</sup></p> |

#### Syntax:

**Sub EA\_OnConnectorTagEdit(Repository As EA.Repository, ConnectorID As Long, String TagName, String TagValue, String TagNotes)**

The *EA\_OnConnectorTagEdit* function syntax contains the following elements:

| Parameter          | Type                                            | Direction | Description                                                                                                                                                                             |
|--------------------|-------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ConnectorID</b> | Long                                            | IN        | The ID of the connector that this Tagged Value is on.                                                                                                                                   |
| <b>Repository</b>  | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p> |
| <b>TagName</b>     | String                                          | IN        | The name of the Tagged Value to edit.                                                                                                                                                   |
| <b>TagNotes</b>    | String                                          | INOUT     | The current value of the Tagged Value notes; if the value is updated, the new value is stored in the repository on exit of the                                                          |

| Parameter       | Type   | Direction | Description                                                                                                               |
|-----------------|--------|-----------|---------------------------------------------------------------------------------------------------------------------------|
|                 |        |           | function.                                                                                                                 |
| <b>TagValue</b> | String | INOUT     | The current value of the tag; if the value is updated, the new value is stored in the repository on exit of the function. |

### 19.3.5.17.3 EA\_OnElementTagEdit

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                   | See also                                                                                                                                                                                                                                                       |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnElementTagEdit</i> is called when the user clicks the ellipsis ( ... ) button for a Tagged Value of type <i>AddinBroadcast</i> on an element.</p> <p>The Add-In displays fields to show and change the value and notes; this function provides the initial values for the Tagged Value notes and value, and takes on any changes on exit of the function.</p> | <p><a href="#">Predefined Structured Types</a> <sup>[1386]</sup></p> <p><a href="#">EA_OnAttributeTagEdit</a> <sup>[2659]</sup></p> <p><a href="#">EA_OnConnectorTagEdit</a> <sup>[2660]</sup></p> <p><a href="#">EA_OnMethodTagEdit</a> <sup>[2662]</sup></p> |

#### Syntax:

**Sub EA\_OnElementTagEdit(Repository As EA.Repository, ObjectID As Long, String TagName, String TagValue, String TagNotes)**

The *EA\_OnElementTagEdit* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                      |
|-------------------|-------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ObjectID</b>   | Long                                            | IN        | The ID of the object (element) that this Tagged Value is on.                                                                                                                     |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |
| <b>TagName</b>    | String                                          | IN        | The name of the Tagged Value to edit.                                                                                                                                            |
| <b>TagNotes</b>   | String                                          | INOUT     | The current value of the Tagged Value notes; if the value is updated, the new value is stored in the repository on exit of the function.                                         |

| Parameter       | Type   | Direction | Description                                                                                                               |
|-----------------|--------|-----------|---------------------------------------------------------------------------------------------------------------------------|
| <b>TagValue</b> | String | INOUT     | The current value of the tag; if the value is updated, the new value is stored in the repository on exit of the function. |

#### 19.3.5.17.4 EA\_OnMethodTagEdit

##### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                             | See also                                                                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnMethodTagEdit</i> is called when the user clicks the ellipsis ( ... ) button for a Tagged Value of type AddinBroadcast on an operation.</p> <p>The Add-In displays fields to show and change the value and notes; this function provides the initial values for the Tagged Value notes and value, and takes on any changes on exit of the function.</p> | <p><a href="#">Predefined Structured Types</a> <sup>[1386]</sup></p> <p><a href="#">EA_OnAttributeTagEdit</a> <sup>[2659]</sup></p> <p><a href="#">EA_OnConnectorTagEdit</a> <sup>[2660]</sup></p> <p><a href="#">EA_OnElementTagEdit</a> <sup>[2661]</sup></p> |

##### Syntax:

**Sub EA\_OnMethodTagEdit(Repository As EA.Repository, MethodID As Long, String TagName, String TagValue, String TagNotes)**

The *EA\_OnMethodTagEdit* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                  |
|-------------------|-------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MethodID</b>   | Long                                            | IN        | The ID of the method that this Tagged Value is on.                                                                                                                           |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br>Poll its members to retrieve model data and user interface status information. |
| <b>TagName</b>    | String                                          | IN        | The name of the Tagged Value to edit.                                                                                                                                        |
| <b>TagNotes</b>   | String                                          | INOUT     | The current value of the Tagged Value notes; if the value is updated, the new value is stored in the repository on exit of the function.                                     |

| Parameter       | Type   | Direction | Description                                                                                                               |
|-----------------|--------|-----------|---------------------------------------------------------------------------------------------------------------------------|
| <b>TagValue</b> | String | INOUT     | The current value of the tag; if the value is updated, the new value is stored in the repository on exit of the function. |

### 19.3.5.18 Technology Events

Enterprise Architect Add-Ins can respond to the following events associated with the use of MDG Technologies:

| Topic                                 | Link                                                          |
|---------------------------------------|---------------------------------------------------------------|
| EA_OnInitializeTechnologies           | <a href="#">EA_OnInitializeTechnologies</a> <sup>[2663]</sup> |
| EA_OnPreActivateTechnology            | <a href="#">EA_OnPreActivateTechnology</a> <sup>[2664]</sup>  |
| EA_OnPostActivateTechnology           | <a href="#">EA_OnPostActivateTechnology</a> <sup>[2665]</sup> |
| EA_OnPreDeleteTechnology (Deprecated) | <a href="#">EA_OnPreDeleteTechnology</a> <sup>[2666]</sup>    |
| EA_OnDeleteTechnology (Deprecated)    | <a href="#">EA_OnDeleteTechnology</a> <sup>[2667]</sup>       |
| EA_OnImportTechnology (Deprecated)    | <a href="#">EA_OnImportTechnology</a> <sup>[2668]</sup>       |

#### 19.3.5.18.1 EA\_OnInitializeTechnologies

##### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                             | See also |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <p><i>EA_OnInitializeTechnologies</i> requests that an Add-In pass an MDG Technology to Enterprise Architect for loading.</p> <p>This event occurs on Enterprise Architect startup. Return your technology XML to this function and Enterprise Architect loads and enables it.</p> |          |

##### Syntax:

**Function** *EA\_OnInitializeTechnologies*(Repository As EA.Repository) As Object

The *EA\_OnInitializeTechnologies* function syntax contains the following element:

| Parameter         | Type                                    | Direction | Description                                                                                                                                                                      |
|-------------------|-----------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b> | <a href="#">EA.Repository</a><br>[2448] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |

**Return Value:**

Return the MDG Technology as a single XML string.

**Example**

```
Public Function EA_OnInitializeTechnologies(ByVal Repository As EA.
Repository) As Object
 EA_OnInitializeTechnologies = My.Resources.MyTechnology
End Function
```

**19.3.5.18.2 EA\_OnPreActivateTechnology****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See also                                                                                                                                                    |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPreActivateTechnology</i> notifies Add-Ins that an MDG Technology resource is about to be activated in the model.</p> <p>This event occurs when a user selects to activate an MDG Technology resource in the model (by clicking on the <b>Set Active</b> button on the MDG Technologies dialog or by selecting the technology in the list box in the Default Tools toolbar).</p> <p>The notification is provided immediately after the user attempts to activate the MDG Technology, so that the Add-In can permit or disable activation of the Technology.</p> | <p><a href="#">MDG Technologies</a> [1285]</p> <p><a href="#">Default Tools Toolbar</a> [139]</p> <p><a href="#">EA_OnPostActivateTechnology</a> [2665]</p> |

**Syntax:**

**Function** *EA\_OnPreActivateTechnology(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreActivateTechnology* function syntax contains the following elements:

| Parameter   | Type                                         | Direction | Description                                                                                                                                                                                        |
|-------------|----------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b> | <a href="#">EA.EventProperties</a><br>[2437] | IN        | Contains the following <i>EventProperty</i> objects for the MDG Technology to be activated: <ul style="list-style-type: none"> <li><i>TechnologyID</i>: A string value corresponding to</li> </ul> |

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                      |
|-------------------|-------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   |                                                 |           | the MDG Technology ID                                                                                                                                                            |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |

**Return Value:**

Return **True** to enable activation of the MDG Technology resource in the model. Return **False** to disable activation of the MDG Technology resource.

**19.3.5.18.3 EA\_OnPostActivateTechnology****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | See also                                                                                                                                                                                 |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPostActivateTechnology</i> notifies Add-Ins that an MDG Technology resource has been activated in the model.</p> <p>This event occurs when a user activates an MDG Technology resource in the model (by clicking on the <b>Set Active</b> button on the MDG Technologies dialog or by selecting the technology in the list box in the Default Tools toolbar).</p> <p>The notification is provided immediately after the user succeeds in activating the MDG Technology, so that the Add-In can update the Technology if necessary.</p> | <p><a href="#">MDG Technologies</a><sup>[1285]</sup></p> <p><a href="#">Default Tools Toolbar</a><sup>[139]</sup></p> <p><a href="#">EA_OnPreActivateTechnology</a><sup>[2664]</sup></p> |

**Syntax:**

**Function** *EA\_OnPostActivateTechnology(Repository As EA.Repository, Info As EA.EventProperties)*

The *EA\_OnPostActivateTechnology* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                                              |
|-------------------|------------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[2431]</sup> | IN        | Contains the following <i>EventProperty</i> objects for the MDG Technology to be activated: <ul style="list-style-type: none"> <li><i>TechnologyID</i>: A string value corresponding to the MDG Technology ID</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open                                                                                                                                                           |

| Parameter | Type | Direction | Description                                                                                                   |
|-----------|------|-----------|---------------------------------------------------------------------------------------------------------------|
|           |      |           | Enterprise Architect model.<br>Poll its members to retrieve model data and user interface status information. |

**Return Value:**

Return **True** if the MDG Technology resource is updated during this notification. Return **False** otherwise.

**19.3.5.18.4 EA\_OnPreDeleteTechnology**

**Deprecated** - refers to deleting a technology through the Resources window; this process is no longer recommended.

See:

| Topic                       | Link                                                                 |
|-----------------------------|----------------------------------------------------------------------|
| EA_OnInitializeTechnologies | <a href="#">EA_OnInitializeTechnologies</a><br><small>[2663]</small> |
| EA_OnPreActivateTechnology  | <a href="#">EA_OnPreActivateTechnology</a><br><small>[2664]</small>  |
| EA_OnPostActivateTechnology | <a href="#">EA_OnPostActivateTechnology</a><br><small>[2665]</small> |

**Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                    | See also                                                       |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPreDeleteTechnology</i> notifies Add-Ins that an MDG Technology resource is about to be deleted from the model.</p> <p>This event occurs when a user deletes an MDG Technology resource from the model.</p> <p>The notification is provided immediately after the user confirms their request to delete the MDG Technology, so that the Add-In can disable deletion of the MDG Technology.</p> | <a href="#">EA_OnDeleteTechnology</a><br><small>[2667]</small> |

**Syntax:**

**Function** *EA\_OnPreDeleteTechnology(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreDeleteTechnology* function syntax contains the following elements:

| Parameter         | Type                                                        | Direction | Description                                                                                                                                                                                                            |
|-------------------|-------------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br><small>[2431]</small> | IN        | Contains the following <i>EventProperty</i> objects for the MDG Technology to be deleted: <ul style="list-style-type: none"> <li><i>TechnologyID</i>: A string value corresponding to the MDG Technology ID</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[2448]</small>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information.                                       |

**Return Value:**

Return **True** to enable deletion of the MDG Technology resource from the model. Return **False** to disable deletion of the MDG Technology resource.

**19.3.5.18.5 EA\_OnDeleteTechnology**

**Deprecated** - refers to deleting a technology through the Resources window; this process is no longer recommended.

See:

| Topic                       | Link                                                                 |
|-----------------------------|----------------------------------------------------------------------|
| EA_OnInitializeTechnologies | <a href="#">EA_OnInitializeTechnologies</a><br><small>[2663]</small> |
| EA_OnPreActivateTechnology  | <a href="#">EA_OnPreActivateTechnology</a><br><small>[2664]</small>  |
| EA_OnPostActivateTechnology | <a href="#">EA_OnPostActivateTechnology</a><br><small>[2665]</small> |

**Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                          | See also                                                          |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| <b>Details</b> | <i>EA_OnDeleteTechnology</i> notifies Add-Ins that an MDG Technology resource has been deleted from the model.<br><br>This event occurs after a user has deleted an MDG Technology resource from the model. Add-Ins that require an MDG Technology resource to be loaded can catch this event to disable certain functionality. | <a href="#">EA_OnPreDeleteTechnology</a><br><small>[2666]</small> |



**Syntax:****Sub EA\_OnDeleteTechnology(Repository As EA.Repository, Info As EA.EventProperties)**

The *EA\_OnDeleteTechnology* function syntax contains the following elements:

| Parameter         | Type                                         | Direction | Description                                                                                                                                                                       |
|-------------------|----------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br>[2437] | IN        | Contains the following <i>EventProperty</i> objects: <ul style="list-style-type: none"> <li><i>TechnologyID</i>: A string value corresponding to the MDG Technology ID</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br>[2448]      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information.  |

**Return Value:**

None.

**19.3.5.18.6 EA\_OnImportTechnology**

**Deprecated** - refers to deleting a technology through the Resources window; this process is no longer recommended.

See:

| Topic                       | Link                                                  |
|-----------------------------|-------------------------------------------------------|
| EA_OnInitializeTechnologies | <a href="#">EA_OnInitializeTechnologies</a><br>[2663] |
| EA_OnPreActivateTechnology  | <a href="#">EA_OnPreActivateTechnology</a><br>[2664]  |
| EA_OnPostActivateTechnology | <a href="#">EA_OnPostActivateTechnology</a><br>[2665] |

**Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                 | See also |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <p><i>EA_OnImportTechnology</i> notifies Add-Ins that you have imported an MDG Technology resource into the model.</p> <p>This event occurs after you have imported an MDG Technology resource into the model. Add-Ins that require an MDG Technology resource to be loaded can catch this Add-In to enable certain functionality.</p> |          |

**Syntax:****Sub *EA\_OnImportTechnology(Repository As EA.Repository, Info As EA.EventProperties)***

The *EA\_OnImportTechnology* function syntax contains the following elements:

| Parameter         | Type                                                        | Direction | Description                                                                                                                                                                       |
|-------------------|-------------------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br><small>[2431]</small> | IN        | Contains the following <i>EventProperty</i> objects: <ul style="list-style-type: none"> <li><i>TechnologyID</i>: A string value corresponding to the MDG Technology ID</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[2448]</small>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br>Poll its members to retrieve model data and user interface status information.      |

**Return Value:**

None.

**19.3.6 Custom Views****Topics**

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See also                                                        |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
| <b>General Usage</b> | <p>Enterprise Architect enables custom windows to be inserted as tabs in the Diagram View that appears at the center of the Enterprise Architect frame.</p> <p><b>Creating a custom view</b> enables you to easily and quickly tab between a custom interface and diagrams and other views normally provided by Enterprise Architect.</p> <p>Uses for this facility include:</p> <ul style="list-style-type: none"> <li>Reports and graphs showing summary data of the model</li> </ul> | <a href="#">Creating a custom view</a><br><small>[2670]</small> |

| Topic | Detail                                                                                                                                                                                                              | See also |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <ul style="list-style-type: none"> <li>• Alternative views of a diagram</li> <li>• Alternative views of the model</li> <li>• Views of external data related to model data</li> <li>• Documentation tools</li> </ul> |          |

### 19.3.6.1 Create a Custom View

#### Topics

| Topic                              | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | See also                                               |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| <b>How to create a Custom View</b> | <p>A custom view must be designed as an ActiveX custom control and inserted through the automation interface.</p> <p>ActiveX custom controls can be created using most well-known programming tools including Microsoft Visual Studio.NET. See the documentation provided by the relevant vendor on how to create a custom control to produce an OCX file.</p> <p>Once the custom control has been created and registered on the target system, it can be added through the <i>AddTab()</i> method of the <i>Repository</i> object.</p> <p>While it is possible to call <i>AddTab()</i> from any automation client, it is likely that you would call it from an Add-In, and that Add-In is defined in the same OCX that provides the custom view.</p> <p>Example C# code is shown below:</p> <pre> public class AddIn {     UserControl1 m_MyControl;      public void EA_Connect ( EA. Repository Rep)     {     }      public object EA_GetMenuItems( EA. Repository Repository, string Location, string MenuName)     {         if( MenuName == "" )             return "- &amp;C# Control Demo";         else         {             String( ) ret = {"&amp;Create", "&amp;Show Button"};             return ret;         }     }      public void EA_MenuClick( EA. Repository Rep, string Location, string MenuName, string ItemName) </pre> | <a href="#">Repository Class</a> <small>[2448]</small> |

| Topic | Detail                                                                                                                                                                                                                                                        | See also |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <pre>         {             if ( ItemName == "&amp;Create" )                 m_MyControl = ( User Control 1) Rep. AddTab( " C# Demo", " Cont Demo. User Control 1" );             else                 m_MyControl . ShowBut t on( ) ;         }     } </pre> |          |

### 19.3.7 Custom Docked Window

#### Topics

| Topic           | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See also                                           |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| <b>Abstract</b> | <p>Enterprise Architect enables a custom docked window to be add to Enterprise Architect and shown and docked in the same way as other Enterprise Architect docked windows.</p> <p>A custom docked window must be designed as an ActiveX custom control and inserted through the automation interface.</p> <p>ActiveX custom controls can be created using most well-known programming tools including Microsoft Visual Studio.NET. See the documentation provided by the relevant vendor on how to create a custom control to produce an OCX file.</p> <p>Once the custom control has been created and registered on the target system, it can be added through the <i>AddWindow()</i> method of the <i>Repository</i> object.</p> <p>While it is possible to call <i>AddWindow()</i> from any automation client, it is likely that you would call it from an Add-In, and that Add-In is defined in the same OCX that provides the custom view.</p> | <a href="#">Repository Class</a> <sup>[2448]</sup> |

### 19.3.8 MDG Add-Ins

#### Topics

| Topic           | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See also                                                                                                                                                                                |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Abstract</b> | <p>MDG Add-Ins are specialized types of Add-Ins that have additional features and extra requirements for Add-In authors who want to contribute to Enterprise Architect's goal of Model Driven Generation.</p> <p>Unlike general Add-In events, MDG Add-In events are only sent to the Add-In that has taken ownership of an Enterprise Architect model branch on a particular workstation.</p> <p>One of the additional responsibilities of an MDG Add-In is to take ownership of a branch of an Enterprise Architect model, which is done through the <i>MDG_Connect</i> event.</p> | <a href="#">MDG_Connect</a> <sup>[2674]</sup><br><a href="#">EA_Connect</a> <sup>[2597]</sup><br><a href="#">MDG Events</a> <sup>[2672]</sup><br><a href="#">Sparx Systems</a> (Online) |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                         | See also  |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
|       | <p>MDG Add-Ins identify themselves as such during <i>EA_Connect</i> by returning the string <b>MDG</b>.</p> <p>Unlike ordinary Add-Ins, responding to <i>MDG Add-In</i> events is not optional, and methods must be published for each of the <i>MDG Events</i>.</p> <p>Two examples of MDG Add-Ins are the commercially available MDG Link for Eclipse and MDG Link for Visual Studio, published by <i>Sparx Systems</i>.</p> | Resource) |

### 19.3.8.1 MDG Events

An MDG Add-In must respond to all MDG Events. These events usually identify processes such as Build, Run, Synchronize, PreMerge and PostMerge, amongst others.

An MDG Link Add-In is expected to implement some form of forward and reverse engineering capability within Enterprise Architect, and as such requires access to a specific set of events, all to do with generation, synchronization and general processes concerned with converting models to code and code to models.

| Topic                    | Link                                                       |
|--------------------------|------------------------------------------------------------|
| MDG_BuildProject         | <a href="#">MDG_BuildProject</a> <sup>[2673]</sup>         |
| MDG_Connect              | <a href="#">MDG_Connect</a> <sup>[2674]</sup>              |
| MDG_Disconnect           | <a href="#">MDG_Disconnect</a> <sup>[2675]</sup>           |
| MDG_GetConnectedPackages | <a href="#">MDG_GetConnectedPackages</a> <sup>[2676]</sup> |
| MDG_GetProperty          | <a href="#">MDG_GetProperty</a> <sup>[2676]</sup>          |
| MDG_Merge                | <a href="#">MDG_Merge</a> <sup>[2678]</sup>                |
| MDG_NewClass             | <a href="#">MDG_NewClass</a> <sup>[2680]</sup>             |
| MDG_PostGenerate         | <a href="#">MDG_PostGenerate</a> <sup>[2681]</sup>         |
| MDG_PostMerge            | <a href="#">MDG_PostMerge</a> <sup>[2682]</sup>            |
| MDG_PreGenerate          | <a href="#">MDG_PreGenerate</a> <sup>[2683]</sup>          |
| MDG_PreMerge             | <a href="#">MDG_PreMerge</a> <sup>[2683]</sup>             |

| Topic          | Link                                                 |
|----------------|------------------------------------------------------|
| MDG_PreReverse | <a href="#">MDG_PreReverse</a> <small>[2684]</small> |
| MDG_RunExe     | <a href="#">MDG_RunExe</a> <small>[2685]</small>     |
| MDG_View       | <a href="#">MDG_View</a> <small>[2686]</small>       |

### 19.3.8.1.1 MDGBuild Project

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                   | See also                                         |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| <b>Details</b> | <p><i>MDG_BuildProject</i> enables the Add-In to handle file changes caused by generation. This function is called in response to a user selecting the <b>Extensions   Build Project</b> menu option.</p> <p>Respond to this event by compiling the project source files into a running application.</p> | <a href="#">MDG_RunExe</a> <small>[2685]</small> |

#### Syntax:

#### *Sub MDG\_BuildProject(Repository As EA.Repository, PackageGuid As String)*

The *MDG\_BuildProject* function syntax contains the following elements:

| Parameter          | Type                                                | Direction | Description                                                                                                                                                                             |
|--------------------|-----------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PackageGuid</b> | <i>String</i>                                       | IN        | The GUID identifying the Enterprise Architect package subtree that is controlled by the Add-In.                                                                                         |
| <b>Repository</b>  | <a href="#">EA.Repository</a> <small>[2448]</small> | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p> |

#### Return Value:

None.

### 19.3.8.1.2 MDGConnect

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | See also                                                |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| <b>Details</b> | <p><i>MDG_Connect</i> enables the Add-In to handle a user driven request to connect a model branch to an external application. This function is called when the user attempts to connect a particular Enterprise Architect package to an as yet unspecified external project. This event enables the Add-In to interact with the user to specify such a project.</p> <p>The Add-In is responsible for retaining the connection details, which should be stored on a per-user or per-workstation basis. That is, users who share a common Enterprise Architect model over a network should be able to connect and disconnect to external projects independently of one another.</p> <p>The Add-In should therefore not store connection details in an Enterprise Architect repository. A suitable place to store such details would be:</p> <pre>SHGet Fo l der Pat h ( . . CSI DL_APPDATA. . ) \ Addi nName</pre> <p>The <i>PackageGuid</i> parameter is the same identifier as is required for most events relating to the MDG Add-In. Therefore it is recommended that the connection details be indexed using the <i>PackageGuid</i> value.</p> <p>The <i>PackageID</i> parameter is provided to aid fast retrieval of package details from Enterprise Architect, should this be required.</p> | <a href="#">MDG_Disconnect</a><br><small>[2675]</small> |

#### Syntax:

**Function** *MDG\_Connect*(*Repository As EA.Repository, PackageID as Long, PackageGuid As String*) *As Long*

The *MDG\_Connect* function syntax contains the following elements:

| Parameter          | Type                          | Direction | Description                                                                                                                                                |
|--------------------|-------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PackageGuid</b> | <i>String</i>                 | IN        | The unique ID identifying the project provided by the Add-In when a connection to a project branch of an Enterprise Architect model was first established. |
| <b>PackageID</b>   | <i>Long</i>                   | IN        | The <i>PackageID</i> of the Enterprise Architect package the user has requested to have connected to an external project.                                  |
| <b>Repository</b>  | <a href="#">EA.Repository</a> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.                                                                 |

| Parameter | Type                   | Direction | Description                                                                    |
|-----------|------------------------|-----------|--------------------------------------------------------------------------------|
|           | <a href="#">[2448]</a> |           | Poll its members to retrieve model data and user interface status information. |

**Return Value:**

Returns a non-zero to indicate that a connection has been made; a zero indicates that the user has not nominated a project and connection should not proceed.

**19.3.8.1.3 MDGDisconnect****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                   | See also                                           |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| <b>Details</b> | <p><i>MDG_Disconnect</i> enables the Add-In to respond to user requests to disconnect the model branch from an external project.</p> <p>This function is called when the user attempts to disconnect an associated external project. The Add-In is required to delete the details of the connection.</p> | <a href="#">MDG_Connect</a> <a href="#">[2674]</a> |

**Syntax:**

**Function** *MDG\_Disconnect(Repository As EA.Repository, PackageGuid As String) As Long*

The *MDG\_Disconnect* function syntax contains the following elements:

| Parameter          | Type                                                 | Direction | Description                                                                                                                                                                             |
|--------------------|------------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PackageGuid</b> | <i>String</i>                                        | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                                        |
| <b>Repository</b>  | <a href="#">EA.Repository</a> <a href="#">[2448]</a> | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p> |

**Return Value:**

Returns a non-zero to indicate that a disconnection has occurred enabling Enterprise Architect to update the user interface. A zero indicates that the user has not disconnected from an external project.



### 19.3.8.1.4 MDGGetConnectedPackages

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                         | See also                                      |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| <b>Details</b> | <p><i>MDG_GetConnectedPackages</i> enables the Add-In to return a list of current connection between Enterprise Architect and an external application.</p> <p>This function is called when the Add-In is first loaded, and is expected to return a list of the available connections to external projects for this Add-In.</p> | <a href="#">MDG_Connect</a> <sup>[2674]</sup> |

#### Syntax:

**Function** *MDG\_GetConnectedPackages(Repository As EA.Repository) As Variant*

The *MDG\_GetConnectedPackages* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                             |
|-------------------|-------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[2448]</sup> | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p> |

#### Return Value:

Returns an array of GUID strings representing individual Enterprise Architect packages.

### 19.3.8.1.5 MDGGetProperty

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <p><i>MDG_GetProperty</i> provides miscellaneous Add-In details to Enterprise Architect.</p> <p>This function is called by Enterprise Architect to poll the Add-In for information relating to the <i>PropertyName</i>. This event should occur in as short a duration as possible as Enterprise Architect does not cache the information provided by the function.</p> <p>Values corresponding to the following <i>PropertyNames</i> must be provided:</p> <ul style="list-style-type: none"> <li><b>IconID</b> - Return the name of a DLL and a resource identifier in the format <i>#ResID</i>, where the resource ID</li> </ul> |          |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>indicates an Icon</p> <p><i>c:\program files\myapp\myapp.dll#101</i></p> <ul style="list-style-type: none"> <li>• <b>Language</b> - Return the default language that Classes should be assigned when they are created in Enterprise Architect</li> <li>• <b>HiddenMenus</b> - Return one or more values from the <i>MDGMenus</i> enumeration to hide menus that do not apply to your Add-In.</li> </ul> <pre>if ( PropertyName == " HiddenMenus" )     return mgBuildProject + mgRun;</pre> |          |

**Syntax:**

**Function** *MDG\_GetProperty(Repository As EA.Repository, PackageGuid As String, PropertyName As String) As Variant*

The *MDG\_GetProperty* function syntax contains the following elements:

| Parameter           | Type                                                   | Direction | Description                                                                                                                                                                      |
|---------------------|--------------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PackageGuid</b>  | <i>String</i>                                          | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                                 |
| <b>PropertyName</b> | <i>String</i>                                          | IN        | The name of the property that is used by Enterprise Architect. See <i>Details</i> for the possible values.                                                                       |
| <b>Repository</b>   | <a href="#">EA.Repository</a><br><small>[2448]</small> | IN        | An <i>EA.Repository</i> object representing the currently-open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |

**Return Value:**

See *Details*, above.

### 19.3.8.1.6 MDGMerge

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See also                                                                                                                                                       |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><i>MDG_Merge</i> enables the Add-In to jointly handle changes to both the model branch and the code project that the model branch is connected to.</p> <p>This event should be called whenever the user has asked to merge their model branch with its connected code project, or whenever the user has established a new connection to a code project.</p> <p>The purpose of this event is to enable the Add-In to interact with the user to perform a merge between the model branch and the connected project.</p> | <p><a href="#">MDG_Connect</a><sup>[2674]</sup></p> <p><a href="#">MDG_PreMerge</a><sup>[2683]</sup></p> <p><a href="#">MDG_PostMerge</a><sup>[2682]</sup></p> |

#### Syntax:

**Function** *MDG\_Merge*(*Repository As EA.Repository, PackageGuid As String, SynchObjects As Variant, SynchType As String, ExportObjects As Variant, ExportFiles As Variant, ImportFiles As Variant, IgnoreLocked As String, Language As String*) *As Long*

The *MDG\_Merge* function syntax contains the following elements:

| Parameter            | Type           | Direction | Description                                                                                                                                                                                                                                                                                                                     |
|----------------------|----------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ExportFiles</b>   | <i>Variant</i> | OUT       | A string array containing the list of files for each model object chosen for export by the Add-In.<br><br>Each entry in this array must have a corresponding entry in the <i>ExportObjects</i> parameter at the same array index, so <i>ExportFiles(2)</i> must contain the filename of the object by <i>ExportObjects(2)</i> . |
| <b>ExportObjects</b> | <i>Variant</i> | OUT       | The string array containing the list of new model objects (in <i>Object ID</i> format) to be exported by Enterprise Architect to the code project.                                                                                                                                                                              |
| <b>IgnoreLocked</b>  | <i>String</i>  | OUT       | A value indicating whether to ignore any files locked by the code project (that is, " <b>TRUE</b> " or " <b>FALSE</b> ").                                                                                                                                                                                                       |
| <b>ImportFiles</b>   | <i>Variant</i> | OUT       | A string array containing the list of code files made available to the code project to be newly imported to the model.<br><br>Enterprise Architect imports each file listed in this array for import into the connected model branch.                                                                                           |
| <b>Language</b>      | <i>String</i>  | OUT       | The string value containing the name of the code language                                                                                                                                                                                                                                                                       |

| Parameter           | Type                                                   | Direction | Description                                                                                                                                                                                    |
|---------------------|--------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     |                                                        |           | supported by the code project connected to the model branch.                                                                                                                                   |
| <b>PackageGuid</b>  | <i>String</i>                                          | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                                               |
| <b>Repository</b>   | <a href="#">EA.Repository</a><br><small>[2448]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information.               |
| <b>SynchObjects</b> | <i>Variant</i>                                         | OUT       | A string array containing a list of objects ( <i>Object ID</i> format) to be jointly synchronized between the model branch and the project.<br><br>See below for the format of the Object IDs. |
| <b>SynchType</b>    | <i>String</i>                                          | OUT       | The value determining the user-selected type of synchronization to take place.<br><br>See below for a list of valid values.                                                                    |

**Return Value:**

Return a non-zero if the merge operation completed successfully and a zero value when the operation has been unsuccessful.

**Merge:**

A merge consists of three major operations:

- **Export:** Where newly created model objects are exported into code and made available to the code project
- **Import:** Where newly created code objects, Classes and such things are imported into the model
- **Synchronize:** Where objects available both to the model and in code are jointly updated to reflect changes made in either the model, code project or both

**Synchronize Type:**

The *Synchronize* operation can take place in one of four different ways. Each of these ways corresponds to a value returned by *SynchType*:

- None: (*SynchType* = 0) No synchronization is to be performed
- Forward: (*SynchType* = 1) Forward synchronization, between the model branch and the code project is to occur
- Reverse: (*SynchType* = 2) Reverse synchronization, between the code project and the model branch is

to occur

- Both: (*SynchType* = 3) Reverse, then Forward synchronization's are to occur

#### **Object ID Format:**

Each of the Object IDs listed in the string arrays described above should be composed in the following format:

( @namespace ) \* ( #class ) \* ( \$attribute | %operation | : property ) \*

#### **19.3.8.1.7 MDGNewClass**

##### **Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                            | See also                                                 |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| <b>Details</b> | <p><i>MDG_NewClass</i> enables the Add-In to alter details of a Class before it is created.</p> <p>This method is called when Enterprise Architect generates a new Class, and requires information relating to assigning the language and file path. The file path should be passed back as a return value and the language should be passed back via the language parameter.</p> | <a href="#">MDG_PreGenerate</a><br><small>[2683]</small> |

##### **Syntax:**

**Function** *MDG\_NewClass*(*Repository As EA.Repository, PackageGuid As String, CodeID As String, Language As String*) *As String*

The *MDG\_NewClass* function syntax contains the following elements:

| Parameter          | Type                                                   | Direction | Description                                                                                                                                                                      |
|--------------------|--------------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CodeID</b>      | <i>String</i>                                          | IN        | A string used to identify the code element before it is created.<br>For more information, see <a href="#">MDG_View</a><br><small>[2686]</small> .                                |
| <b>Language</b>    | <i>String</i>                                          | OUT       | A string used to identify the programming language for the new Class. The language must be supported by Enterprise Architect.                                                    |
| <b>PackageGuid</b> | <i>String</i>                                          | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                                 |
| <b>Repository</b>  | <a href="#">EA.Repository</a><br><small>[2448]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |

**Return Value:**

Returns a string containing the file path that should be assigned to the Class.

**19.3.8.1.8 MDGPostGenerate****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See also                                  |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| <b>Details</b> | <p><i>MDG_PostGenerate</i> enables the Add-In to handle file changes caused by generation.</p> <p>This event is called after Enterprise Architect has prepared text to replace the existing contents of a file. Responding to this event enables the Add-In to write to the linked application's user interface rather than modify the file directly.</p> <p>When the contents of a file are changed, Enterprise Architect passes <i>FileContents</i> as a non-empty string. New files created as a result of code generation are also sent through this mechanism, enabling Add-Ins to add new files to the linked project's file list.</p> <p>When new files are created Enterprise Architect passes <i>FileContents</i> as an empty string. When a non-zero is returned by this function, the Add-In has successfully written the contents of the file. A zero value for the return indicates to Enterprise Architect that the file must be saved.</p> | <a href="#">MDG_PreGenerate</a><br>[2683] |

**Syntax:**

**Function** *MDG\_PostGenerate(Repository As EA.Repository, PackageGuid As String, FilePath As String, FileContents As String) As Long*

The *MDG\_PostGenerate* function syntax contains the following elements:

| Parameter           | Type                                    | Direction | Description                                                                                                                                                                      |
|---------------------|-----------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FileContents</b> | <i>String</i>                           | IN        | A string containing the proposed contents of the file.                                                                                                                           |
| <b>FilePath</b>     | <i>String</i>                           | IN        | The path of the file Enterprise Architect intends to overwrite.                                                                                                                  |
| <b>PackageGuid</b>  | <i>String</i>                           | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                                 |
| <b>Repository</b>   | <a href="#">EA.Repository</a><br>[2448] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |

| Parameter | Type | Direction | Description |
|-----------|------|-----------|-------------|
|           |      |           |             |

**Return Value:**

The return value depends on the type of event that this function is responding to (see *Details*, above). This function is required to handle two separate and distinct cases.

**19.3.8.1.9 MDGPostMerge****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                | See also                                                                                                                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p>MDG_PostMerge is called after a merge process has been completed.</p> <p>This function is called by Enterprise Architect after the merge process has been completed.</p> <p>File save checking should not be performed with this function, but should be handled by <i>MDG_PreGenerate</i>, <i>MDG_PostGenerate</i> and <i>MDG_PreReverse</i>.</p> | <p><a href="#">MDG_PreGenerate</a><br/>[2683]</p> <p><a href="#">MDG_PostGenerate</a><br/>[2681]</p> <p><a href="#">MDG_PreReverse</a><br/>[2684]</p> <p><a href="#">MDG_PreMerge</a><br/>[2683]</p> <p><a href="#">MDG_Merge</a> [2678]</p> |

**Syntax:**

**Function** MDG\_PostMerge(*Repository As EA.Repository, PackageGuid As String*) As Long

The *MDG\_PostMerge* function syntax contains the following elements:

| Parameter          | Type                                    | Direction | Description                                                                                                                                                                             |
|--------------------|-----------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b>  | <a href="#">EA.Repository</a><br>[2448] | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p> |
| <b>PackageGuid</b> | <i>String</i>                           | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                                        |

**Return Value:**

Return a zero value if the post-merge process has failed, a non-zero return indicates that the post-merge has been successful. Enterprise Architect assumes a non-zero return if this method is not implemented

### 19.3.8.1.10 MDGPreGenerate

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                     | See also                                   |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| <b>Details</b> | <p><i>MDG_PreGenerate</i> enables the Add-In to deal with unsaved changes.</p> <p>This function is called immediately before Enterprise Architect attempts to generate files from the model. A possible use of this function would be to prompt the user to save unsaved source files.</p> | <a href="#">MDG_PostGenerate</a><br>[2681] |

#### Syntax:

**Function** *MDG\_PreGenerate(Repository As EA.Repository, PackageGuid As String) As Long*

The *MDG\_PreGenerate* function syntax contains the following elements:

| Parameter          | Type                                    | Direction | Description                                                                                                                                                                             |
|--------------------|-----------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b>  | <a href="#">EA.Repository</a><br>[2448] | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p> |
| <b>PackageGuid</b> | <i>String</i>                           | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                                        |

#### Return Value:

Return a zero value to abort generation. Any other value enables the generation to continue.

### 19.3.8.1.11 MDGPreMerge

#### Topics

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                               | See also                                                                                                                                                                |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><i>MDG_PreMerge</i> is called after a merge process has been initiated by the user and before Enterprise Architect performs the merge process.</p> <p>This event is called after a user has performed their interactions with the merge screen and has confirmed the merge with the <b>OK</b> button, but before Enterprise Architect performs the merge process using the data provided by the <i>MDG_Merge</i> call, before</p> | <a href="#">MDG_PreGenerate</a><br>[2683]<br><a href="#">MDG_PostGenerate</a><br>[2681]<br><a href="#">MDG_PreReverse</a><br>[2684]<br><a href="#">MDG_Merge</a> [2678] |



| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See also                                |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
|       | <p>any changes have been made to the model or the connected project.</p> <p>This event is made available to provide the Add-In with the opportunity to generally set internal Add-In flags to augment the <i>MDG_PreGenerate</i>, <i>MDG_PostGenerate</i> and <i>MDG_PreReverse</i> events.</p> <p>File save checking should not be performed with this function, but should be handled by <i>MDG_PreGenerate</i>, <i>MDG_PostGenerate</i> and <i>MDG_PreReverse</i>.</p> | <a href="#">MDG_PostMerge</a><br>[2682] |

**Syntax:****Function** *MDG\_PreMerge(Repository As EA.Repository, PackageGuid As String) As Long*

The *MDG\_PreMerge* function syntax contains the following elements:

| Parameter          | Type                                    | Direction | Description                                                                                                                                                                             |
|--------------------|-----------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b>  | <a href="#">EA.Repository</a><br>[2448] | IN        | <p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p> |
| <b>PackageGuid</b> | <i>String</i>                           | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                                        |

**Return Value:**

A return value of zero indicates that the merge process can not occur. If the value is not zero the merge process proceeds.

If this method is not implemented then it is assumed that a merge process is used.

**19.3.8.1.12 MDGPreReverse****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                    | See also                                                                                    |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><i>MDG_PreReverse</i> enables the Add-In to save file changes before being imported into Enterprise Architect.</p> <p>This function operates on a list of files that are about to be reverse-engineered into Enterprise Architect. If the user is working on unsaved versions of these files in an editor, you could either prompt the user or save automatically.</p> | <a href="#">MDG_PostGenerate</a><br>[2681]<br><br><a href="#">MDG_PreGenerate</a><br>[2683] |

| Topic | Detail | See also |
|-------|--------|----------|
|       |        |          |

**Syntax:**

**Sub MDG\_PreReverse(Repository As EA.Repository, PackageGuid As String, FilePaths As Variant)**

The *MDG\_PreReverse* function syntax contains the following elements:

| Parameter          | Type                                                   | Direction | Description                                                                                                                                                                      |
|--------------------|--------------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b>  | <a href="#">EA.Repository</a><br><small>[2448]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |
| <b>PackageGuid</b> | <i>String</i>                                          | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                                 |
| <b>FilePaths</b>   | <i>String array</i>                                    | IN        | An array of filepaths pointed to the files that are to be reverse engineered.                                                                                                    |

**Return Value:**

None.

**19.3.8.1.13 MDGRunExe****Topics**

| Topic          | Detail                                                                                                                                                                                                                                    | See also                                                  |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| <b>Details</b> | <i>MDG_RunExe</i> enables the Add-In to run the target application.<br><br>This function is called when the user selects the <b>Extensions   Run Exe</b> menu option.<br><br>Respond to this event by launching the compiled application. | <a href="#">MDG_BuildProject</a><br><small>[2673]</small> |

**Syntax:**

**Sub MDG\_RunExe(Repository As EA.Repository, PackageGuid As String)**

The *MDG\_RunExe* function syntax contains the following elements:

| Parameter          | Type                                    | Direction | Description                                                                                                                                                                      |
|--------------------|-----------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b>  | <a href="#">EA.Repository</a><br>[2448] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |
| <b>PackageGuid</b> | <i>String</i>                           | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                                 |

**Return Value:**

None.

**19.3.8.1.14 MDGView****Topics**

| Topic          | Detail                                                                                                                                                                                                                                                                             | See also |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <i>MDG_View</i> enables the Add-In to display user specified code elements.<br><br>This function is called by Enterprise Architect when the user asks to view a particular code element. This enables the Add-In to present that element in its own way, usually in a code editor. |          |

**Syntax:*****Function MDG\_View(Repository As EA.Repository, PackageGuid As String, CodeID as String) As Long***The *MDG\_View* function syntax contains the following elements:

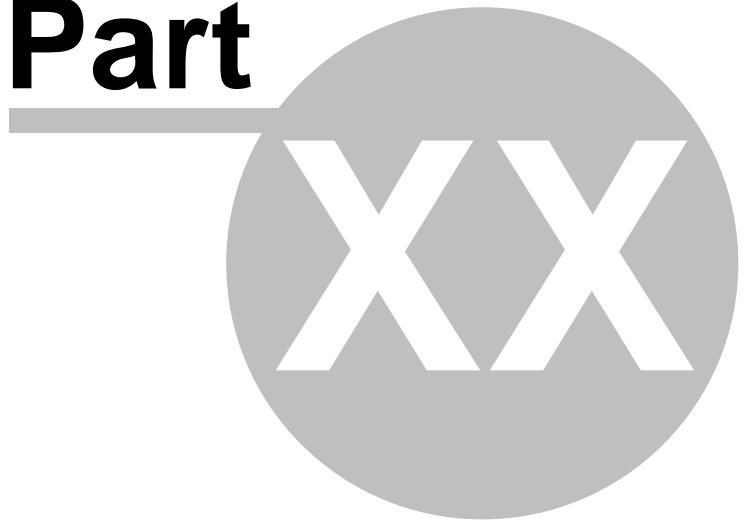
| Parameter          | Type                 | Direction | Description                                                                                                                                                                      |
|--------------------|----------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b>  | <i>EA.Repository</i> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.<br><br>Poll its members to retrieve model data and user interface status information. |
| <b>PackageGuid</b> | <i>String</i>        | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                                 |
| <b>CodeID</b>      | <i>String</i>        | IN        | Identifies the code element in the following format:<br><br><type>ElementPart<type>ElementPart...                                                                                |

| Parameter | Type | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           |      |           | <p>where each element is preceded with a token identifying its type:</p> <ul style="list-style-type: none"><li>@ -namespace</li><li># - Class</li><li>\$ - attribute</li><li>% - operation</li></ul> <p>For example if a user has selected the <i>m_Name</i> attribute of <i>Class1</i> located in <i>namespace Name1</i>, the <i>class ID</i> would be passed through in the following format:</p> <pre>@Name1#Class1%m_Name</pre> |

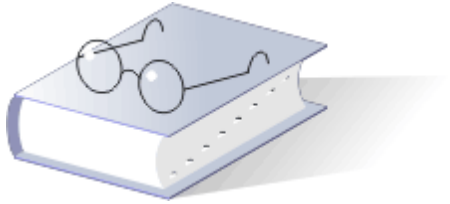
**Return Value:**

Return a non-zero value to indicate that the Add-In has processed the request. Returning a zero value results in Enterprise Architect employing the standard viewing process which is to launch the associated source file.

**Part**



## 20 Glossary



This topic provides a detailed glossary for Enterprise Architect.

[A](#)  
2690

[B](#)  
2693

[C](#)  
2695

[D](#)  
2699

[E](#)  
2702

[F](#)  
2704

[G](#)  
2706

[H](#)  
2707

[I](#)  
2708

[J](#)  
2710

[L](#)  
2711

[M](#)  
2712

[N](#)  
2715

[O](#)  
2716

[P](#)  
2717

[Q](#)  
2721

[R](#)  
2722

[S](#)  
2725

[T](#)  
2730

[U](#)  
2732

[V](#)  
2734

## 20.1 A

| Name                    | Detail                                                                                                                                                                                                                                                                            | See also             |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| <b>Abstract Class</b>   | A Class that cannot be directly instantiated.                                                                                                                                                                                                                                     | Concrete Class       |
| <b>Abstraction</b>      | The essential characteristics of an entity that distinguish it from all other kinds of entities. An abstraction defines a boundary relative to the perspective of the viewer.                                                                                                     |                      |
| <b>Action</b>           | The specification of an executable statement that forms an abstraction of a computation procedure. An action typically results in a change in the state of the system, and can be realized by sending a message to an object or modifying a connector or a value of an attribute. |                      |
| <b>Action Sequence</b>  | An expression that resolves to a sequence of actions.                                                                                                                                                                                                                             |                      |
| <b>Action State</b>     | A state that represents the execution of an atomic action, typically the invocation of an operation.                                                                                                                                                                              |                      |
| <b>Activation</b>       | The execution of an action.                                                                                                                                                                                                                                                       |                      |
| <b>Active Class</b>     | A Class whose instances are active objects. When instantiated, an active Class controls its execution. Rather than being invoked or activated by other objects, it can operate standalone, and define its own thread of behavior.                                                 | Active Object        |
| <b>Active Object</b>    | An object that owns a thread and can initiate control activity. An instance of active Class.                                                                                                                                                                                      | Active Class, Thread |
| <b>Activity</b>         | Defines the bounds for the structural organization that contains a set of basic or fundamental behaviors. It can be used to model procedural type application development for system design through to modeling business processes in organizational structures and work flow.    |                      |
| <b>Activity Diagram</b> | A diagram used to model procedural type application development for system design through to modeling business processes in organizational structures and work flow.                                                                                                              |                      |
| <b>Activity Graph</b>   | A special case of a State Machine that is used to model processes involving one or more classifiers.                                                                                                                                                                              | State Chart Diagram  |

|                            |                                                                                                                                                                                                                                                                                                                         |                                                    |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| <b>Actor ( Class )</b>     | A coherent set of roles that users of Use Cases play when interacting with these Use Cases. An Actor has one role for each Use Case with which it communicates.                                                                                                                                                         |                                                    |
| <b>Actual Parameter</b>    | A binding for a parameter that resolves to a run-time instance.                                                                                                                                                                                                                                                         | Argument, (Formal) Parameter                       |
| <b>Aggregate ( Class )</b> | A Class that represents the 'whole' in an Aggregation (whole-part) relationship.                                                                                                                                                                                                                                        | Aggregation                                        |
| <b>Aggregation</b>         | A special form of Association that specifies a whole-part relationship between the Aggregate (whole) and a component part.                                                                                                                                                                                              | Composition                                        |
| <b>Analysis</b>            | The part of the software development process whose primary purpose is to formulate a model of the problem domain. Analysis focuses on what to do, design focuses on how to do it.                                                                                                                                       | Design                                             |
| <b>Analysis Diagram</b>    | A diagram used to capture high level business processes and early models of system behavior and elements. It is less formal than some other diagrams, but provides a good means of capturing the essential business characteristics and requirements.                                                                   |                                                    |
| <b>Analysis Time</b>       | Refers to something that occurs during an analysis phase of the software development process.                                                                                                                                                                                                                           | Modeling Time, Run Time, Compile Time, Design Time |
| <b>Architecture</b>        | The organizational structure and associated behavior of a system. An architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts. Parts that interact through interfaces include Classes, Components and subsystems. |                                                    |
| <b>Argument</b>            | A binding for a parameter that resolves to a run-time instance.                                                                                                                                                                                                                                                         | Actual Parameter (Formal) Parameter                |
| <b>Artifact</b>            | A physical piece of information that is used or produced by a business or development process. Examples of Artifacts include models, source files, scripts, and binary executable files. An Artifact can constitute the implementation of a deployable component.                                                       | Product, Component                                 |
| <b>Assembly</b>            | A connector that bridges the required interface of a                                                                                                                                                                                                                                                                    |                                                    |



|                          |                                                                                                                                                                                                                                                                                                                                |                         |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
|                          | component with the provided interface of a second component.                                                                                                                                                                                                                                                                   |                         |
| <b>Association</b>       | The semantic relationship between two or more classifiers that specifies connections among their instances.                                                                                                                                                                                                                    | Link                    |
| <b>Association Class</b> | A model element that has both Association and Class properties. An Association Class can be seen as an Association that also has Class properties, or as a Class that also has Association properties.                                                                                                                         | Class                   |
| <b>Association End</b>   | The endpoint of an Association, which connects the Association to a classifier.                                                                                                                                                                                                                                                | Classifier,<br>Link End |
| <b>Attribute</b>         | A feature within a classifier that describes a range of values that instances of the classifier can hold.                                                                                                                                                                                                                      |                         |
| <b>Auxiliary Class</b>   | A stereotyped Class that supports another more central or fundamental Class, typically by implementing secondary logic or control flow. Auxiliary Classes are typically used together with focus Classes, and are particularly useful for specifying the secondary business logic or control flow of components during design. | Focus Class             |

## 20.2 B

| Name                           | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also          |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <b>Behavior</b>                | The observable effects of an operation or event, including its results.                                                                                                                                                                                                                                                                                                                                                                                                     |                   |
| <b>Behavioral Diagram</b>      | A diagram that depicts the behavioral features of a system or business process.<br><br>Behavioral diagrams include Activity diagrams, State Machine diagrams, Communication diagrams, Interaction Overview diagrams, Sequence diagrams, Timing diagrams and Use Case diagrams.                                                                                                                                                                                              |                   |
| <b>Behavioral Feature</b>      | A dynamic feature of a model element, such as an operation or method.                                                                                                                                                                                                                                                                                                                                                                                                       |                   |
| <b>Behavioral Model aspect</b> | A model aspect that emphasizes the behavior of the instances in a system, including their methods, collaborations, and state histories.                                                                                                                                                                                                                                                                                                                                     |                   |
| <b>Binary Association</b>      | An Association between two Classes; a special case of an N-ary Association.                                                                                                                                                                                                                                                                                                                                                                                                 | N-ary Association |
| <b>Binding</b>                 | The creation of a model element from a template by supplying arguments for the parameters of the template.                                                                                                                                                                                                                                                                                                                                                                  |                   |
| <b>Bookmark</b>                | A marker in a Rich Text Format document that enables you to link inner sections of a document into a master document (using the Word 'Insert File' function).                                                                                                                                                                                                                                                                                                               |                   |
| <b>Boolean</b>                 | An enumeration whose values are <b>true</b> and <b>false</b> .                                                                                                                                                                                                                                                                                                                                                                                                              |                   |
| <b>Boolean Expression</b>      | An expression that evaluates to a boolean value.                                                                                                                                                                                                                                                                                                                                                                                                                            |                   |
| <b>Boundary</b>                | <ol style="list-style-type: none"> <li>1. A stereotyped Class that models some system boundary – typically a user interface screen; it is: <ul style="list-style-type: none"> <li>• Used in the conceptual phase to capture user interaction with the system at a screen level (or some other boundary interface type)</li> <li>• Often used in Sequence and Robustness (Analysis) diagrams</li> <li>• The View in the Model-View-Controller pattern</li> </ul> </li> </ol> |                   |

---

| Name | Detail                                                                          | See also |
|------|---------------------------------------------------------------------------------|----------|
|      | 2. A System Boundary element used to delineate a particular part of the system. |          |

## 20.3 C

| Name                  | Detail                                                                                                                                                                                                                               | See also                                                                   |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| <b>C++</b>            | An object-oriented programming language based on the earlier 'C' language.                                                                                                                                                           |                                                                            |
| <b>Call</b>           | An action state that invokes an operation on a classifier.                                                                                                                                                                           |                                                                            |
| <b>Cardinality</b>    | The number of elements in a set.                                                                                                                                                                                                     | Multiplicity                                                               |
| <b>CASE</b>           | Computer Aided Software Engineering. A tool designed for the purpose of modeling and building software systems.                                                                                                                      |                                                                            |
| <b>Child</b>          | In a Generalization relationship, the specialization of another element, the parent.                                                                                                                                                 | Subclass<br>Subtype<br>Parent                                              |
| <b>Choice</b>         | A pseudo-state used to compose complex transitional paths, where the outgoing transition path is decided by dynamic, run-time conditions determined by the actions performed by the State Machine on the path leading to the choice. |                                                                            |
| <b>Class</b>          | A description of a set of objects that share the same attributes, operations, methods, relationships and semantics. A Class can use a set of interfaces to specify collections of operations it provides to its environment.         | Interface<br>Object                                                        |
| <b>Class Diagram</b>  | A diagram that shows a collection of declarative (static) model elements, such as Classes, types, and their contents and relationships.                                                                                              | Object Diagram                                                             |
| <b>Classification</b> | The assignment of an object to a classifier.                                                                                                                                                                                         | Dynamic Classification<br>Multiple Classification<br>Static Classification |
| <b>Classifier</b>     | A mechanism that describes behavioral and structural features. Classifiers include Interfaces, Classes, datatypes, and components.                                                                                                   |                                                                            |
| <b>Client</b>         | A classifier that requests a service from another classifier.                                                                                                                                                                        | Supplier                                                                   |

| Name                         | Detail                                                                                                                                                                                                                                                                                                                                     | See also                                            |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| <b>Collaboration</b>         | The specification of how an operation or classifier, such as a Use Case, is realized by a set of classifiers and Associations playing specific roles used in a specific way. The Collaboration defines an interaction.                                                                                                                     | Interaction                                         |
| <b>Collaboration Diagram</b> | Used pre-UML 2.0. Now called a Communication diagram.                                                                                                                                                                                                                                                                                      |                                                     |
| <b>Collaboration Use</b>     | Uses an occurrence to apply a pattern defined by a Collaboration to a specific situation.                                                                                                                                                                                                                                                  |                                                     |
| <b>Combined Fragment</b>     | A combined fragment reflects a piece or pieces of interaction (called interaction operands) controlled by an interaction operator, whose corresponding boolean conditions are known as interaction constraints. It appears graphically as a transparent window, divided by horizontal dashed lines for each operand.                       |                                                     |
| <b>Comment</b>               | An annotation attached to an element or a collection of elements. A comment, or note, has no semantics.                                                                                                                                                                                                                                    | Constraint                                          |
| <b>Communication Diagram</b> | A diagram that shows the interactions between elements at run-time in much the same manner as a Sequence diagram. However, Communication diagrams are used to visualize inter-object relationships, while Sequence diagrams are more effective at visualizing processing over time.                                                        | Collaboration Diagram, Object Diagram               |
| <b>Compile Time</b>          | Refers to something that occurs during the compilation of a software module.                                                                                                                                                                                                                                                               | Modeling Time, Run Time, Analysis Time, Design Time |
| <b>Component</b>             | A modular, deployable, and replaceable part of a system that encapsulates implementation and exposes a set of interfaces. A Component is typically specified by one or more classifiers (such as implementation Classes) that reside on it, and can be implemented by one or more artifacts (such as binary, executable, or script files). | Module Artifact, Product                            |
| <b>Component Diagram</b>     | A diagram that shows the organizations and dependencies among Components.                                                                                                                                                                                                                                                                  |                                                     |
| <b>Composite ( Class )</b>   | A Class that is related to one or more Classes by a Composition relationship.                                                                                                                                                                                                                                                              | Composition                                         |

| Name                               | Detail                                                                                                                                                                                                                                                                                                                                             | See also                                         |
|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| <b>Composite State</b>             | A State that consists of either concurrent (orthogonal) substates or sequential (disjoint) substates.                                                                                                                                                                                                                                              | Substate, Concurrent Substate, Disjoint Substate |
| <b>Composite Structure Diagram</b> | A diagram that reflects the internal collaboration of Classes, Interfaces, or Components to describe a functionality. Composite Structure diagrams are similar to Class diagrams, except that they model a specific usage of the structure.                                                                                                        |                                                  |
| <b>Composition</b>                 | A form of Aggregation that requires that a part instance be included in at most one Composite at a time, and that the Composite object is responsible for the creation and destruction of the parts. Composition can be recursive.                                                                                                                 | Composite Aggregation                            |
| <b>Concrete Class</b>              | A Class that can be directly instantiated.                                                                                                                                                                                                                                                                                                         | Abstract Class                                   |
| <b>Concurrency</b>                 | The occurrence of two or more activities during the same time interval. Concurrency can be achieved by interleaving or simultaneously executing two or more threads.                                                                                                                                                                               | Thread                                           |
| <b>Concurrent Substate</b>         | A substate that can be held simultaneously with other substates contained in the same composite State.                                                                                                                                                                                                                                             | Composite State<br>Disjoint Substate             |
| <b>Connector</b>                   | A logical link between model elements. Can be structural, dynamic or possessive.                                                                                                                                                                                                                                                                   |                                                  |
| <b>Constraint</b>                  | <ol style="list-style-type: none"> <li>1. A semantic condition or restriction. Certain constraints are predefined in the UML, others can be user defined. Constraints are one of three extensibility mechanisms in UML.</li> <li>2. A rule or condition that applies to some element. It is often modeled as a pre- or post- condition.</li> </ol> | Tagged Value, Stereotype<br>Comment              |
| <b>Container</b>                   | <ol style="list-style-type: none"> <li>1. An instance that exists to contain other instances, and that provides operations to access or iterate over its contents.(for example, arrays, lists, sets).</li> <li>2. A component that exists to contain other components.</li> </ol>                                                                  |                                                  |
| <b>Containment Hierarchy</b>       | A namespace hierarchy consisting of model elements, and the containment relationships that exist between them. A containment hierarchy forms a graph.                                                                                                                                                                                              |                                                  |
| <b>Context</b>                     | A view of a set of related modeling elements for a                                                                                                                                                                                                                                                                                                 |                                                  |

| Name                | Detail                                                                                                                                                                                           | See also |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                     | particular purpose, such as specifying an operation.                                                                                                                                             |          |
| <b>Continuation</b> | A Continuation is used in seq and alt combined fragments, to indicate the branches of continuation an operand follows.                                                                           |          |
| <b>Control</b>      | A stereotyped Class that represents a controlling entity or manager. A Control organizes and schedules other activities and elements. It is the controller of the Model-View-Controller pattern. |          |
| <b>Control Flow</b> | A connector linking two nodes in an activity diagram. Control Flow connectors start a node's activity when the preceding node's action is finished.                                              |          |

## 20.4 D

| Name                        | Detail                                                                                                                                                                                                                                                                                  | See also           |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| <b>Database Schema</b>      | The description of a database structure. It defines tables and fields and the relationship between them.                                                                                                                                                                                |                    |
| <b>Datastore</b>            | An element used to define permanently stored data. A token of data that is stored in the Datastore is stored permanently. A token of data that comes out of the Datastore is a copy of the original data. The tokens imported are kept for the life of the Activity in which it exists. |                    |
| <b>Datatype</b>             | A descriptor of a set of values that lack identity and whose operations do not have side effects. Datatypes include primitive pre-defined types and user-definable types. Pre-defined types include numbers, string and time. User-definable types include enumerations.                |                    |
| <b>Decision</b>             | An element of an Activity diagram that indicates a point of conditional progression: if a condition is true, then processing continues one way, if not, then another.                                                                                                                   |                    |
| <b>Defining Model (MOF)</b> | The model on which a repository is based. Any number of repositories can have the same defining model.                                                                                                                                                                                  |                    |
| <b>Delegate</b>             | A connector that defines the internal assembly of a component's external ports and interfaces. Using a Delegate connector wires the internal workings of the system to the outside world, by a delegation of the external interfaces' connections.                                      |                    |
| <b>Delegation</b>           | The ability of an object to issue a message to another object in response to a message. Delegation can be used as an alternative to inheritance.<br><br>Contrast: inheritance                                                                                                           |                    |
| <b>Dependency</b>           | A relationship between two modeling elements, in which a change to one modeling element (the independent element) affects the other modeling element (the dependent element).                                                                                                           |                    |
| <b>Deployment</b>           | A type of Dependency relationship that indicates the deployment of an artifact onto a node or executable target.                                                                                                                                                                        |                    |
| <b>Deployment</b>           | A diagram that shows the configuration of run-time                                                                                                                                                                                                                                      | Component Diagrams |



| Name                            | Detail                                                                                                                                                                                                                                                                                                                         | See also                                                   |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| <b>Diagram</b>                  | processing nodes and the components, processes, and objects that live on them. Components represent run-time manifestations of code units.                                                                                                                                                                                     |                                                            |
| <b>Deployment Specification</b> | Specifies parameters guiding deployment of an artifact, as is common with most hardware and software technologies.                                                                                                                                                                                                             |                                                            |
| <b>Derived Element</b>          | A model element that can be computed from another element, but that is shown for clarity or that is included for design purposes even though it adds no semantic information.                                                                                                                                                  |                                                            |
| <b>Design</b>                   | The part of the software development process whose primary purpose is to decide how the system is to be implemented. During design, strategic and tactical decisions are made to meet the required functional and quality requirements of a system.                                                                            | Analysis                                                   |
| <b>Design Time</b>              | Refers to something that occurs during a design phase of the software development process.                                                                                                                                                                                                                                     | Modeling Time<br>Run Time<br>Compile Time<br>Analysis Time |
| <b>Development Process</b>      | A set of partially ordered steps performed for a given purpose during software development, such as constructing models or implementing models.                                                                                                                                                                                |                                                            |
| <b>Diagram</b>                  | A graphical presentation of a collection of model elements, most often rendered as a connected graph of arcs (relationships) and vertices (other model elements). UML supports 14 diagram types, and Enterprise Architect extends these with seven more. Add-Ins, technologies and profiles can provide further diagram types. |                                                            |
| <b>Diagram Gate</b>             | A simple graphical way to indicate the point at which messages can be transmitted into and out of Interaction Fragments.                                                                                                                                                                                                       |                                                            |
| <b>Diagram View</b>             | The Enterprise Architect workspace area where the UML diagrams are displayed.                                                                                                                                                                                                                                                  |                                                            |
| <b>Disjoint Substate</b>        | A substate that cannot be held simultaneously with other substates contained in the same composite State.                                                                                                                                                                                                                      | Composite State                                            |

| Name                          | Detail                                                                                                                                                                    | See also                                         |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
|                               |                                                                                                                                                                           | Substate<br>Concurrent Substate                  |
| <b>Distribution Unit</b>      | A set of objects or components that are allocated to a process or a processor as a group. A distribution unit can be represented by a run-time composite or an Aggregate. |                                                  |
| <b>Domain</b>                 | An area of knowledge or activity characterized by a set of concepts and terminology understood by practitioners in that area.                                             |                                                  |
| <b>Dynamic Classification</b> | A semantic variation of Generalization in which an object can change its classifier.                                                                                      | Multiple Classification<br>Static Classification |

## 20.5 E

| Name                     | Detail                                                                                                                                                                                                                                               | See also   |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>Element</b>           | <ol style="list-style-type: none"> <li>1. An atomic constituent of a model.</li> <li>2. A model object of any type, such as Class, Component, Node or Object.</li> </ol>                                                                             |            |
| <b>Endpoint</b>          | Used in Interaction diagrams to reflect a lost message in sequence.                                                                                                                                                                                  |            |
| <b>Entity</b>            | A store or persistence mechanism that captures the information or knowledge in a system. It is the Model in the Model-View-Controller pattern.                                                                                                       |            |
| <b>Entry Action</b>      | An action executed upon entering a state in a State Machine regardless of the transition taken to reach that state.                                                                                                                                  |            |
| <b>Entry Point</b>       | Used to define where external states can enter a Sub Machine.                                                                                                                                                                                        |            |
| <b>Enumeration</b>       | A list of named values used as the range of a particular attribute type. For example, RGBColor = {red, green, blue}. Boolean is a predefined enumeration with values from the set {false, true}.                                                     |            |
| <b>Event</b>             | The specification of a significant occurrence that has a location in time and space. In the context of State diagrams, an event is an occurrence that can trigger a transition.                                                                      |            |
| <b>Exception Handler</b> | An element that defines the group of operations to carry out when an exception occurs.                                                                                                                                                               |            |
| <b>Exit Action</b>       | An action executed upon exiting a State in a State Machine regardless of the transition taken to exit that State.                                                                                                                                    |            |
| <b>Exit Point</b>        | Used in Sub Machine states and State Machines to denote the point where the machine is exited and the transition sourcing this exit point, for Sub Machines, is triggered. Exit points are a type of pseudo-state used in the State Machine diagram. |            |
| <b>Export</b>            | In the context of packages, to make an element visible outside its enclosing namespace.                                                                                                                                                              | Visibility |

| Name                    | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also              |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Import                |
| <b>Expose Interface</b> | A toolbox icon that is a graphical way to depict the required and supplied interfaces of a Component, Class or Part.                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                       |
| <b>Expression</b>       | <p>A string that evaluates to a value of a particular type. For example, the expression <math>(7 + 5 * 3)</math> evaluates to a value of type number.</p> <p>A relationship from an extension Use Case to a base Use Case, specifying how the behavior defined for the extension Use Case augments (subject to conditions specified in the extension) the behavior defined for the base Use Case. The behavior is inserted at the location defined by the extension point in the base Use Case. The base Use Case does not depend on performing the behavior of the extension Use Case.</p> | Extend<br>Include     |
| <b>Extend</b>           | A connector used to indicate that an element extends the behavior of another. Extensions are used in Use Case models to indicate one Use Case (optionally) extends the behavior of another.                                                                                                                                                                                                                                                                                                                                                                                                 | Expression<br>Include |

## 20.6 F

| Name                       | Detail                                                                                                                                                                                                                                                                                                                  | See also        |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <b>Facade</b>              | A stereotyped package containing only references to model elements owned by another package; it is used to provide a 'public view' of some of the contents of a package.                                                                                                                                                |                 |
| <b>Feature</b>             | A property, like an operation or attribute, that is encapsulated within a classifier such as an Interface, Class, or Datatype.                                                                                                                                                                                          |                 |
| <b>Final</b>               | A pseudo-state that indicates an end.                                                                                                                                                                                                                                                                                   |                 |
| <b>Final State</b>         | A special kind of State signifying that the enclosing composite State or the entire State Machine is completed.                                                                                                                                                                                                         |                 |
| <b>Fire</b>                | To execute a State transition.                                                                                                                                                                                                                                                                                          | Transition      |
| <b>Flow Final</b>          | An element that depicts an exit from the system, as opposed to the Activity Final, which represents the completion of the activity.                                                                                                                                                                                     |                 |
| <b>Focus Class</b>         | A stereotyped Class that defines the core logic or control flow for one or more auxiliary Classes that support it.<br><br>Focus Classes are typically used together with one or more auxiliary Classes, and are particularly useful for specifying the core business logic or control flow of components during design. | Auxiliary Class |
| <b>Focus of Control</b>    | A symbol on a Sequence diagram that shows the period of time during which an object is performing an action, either directly or through a subordinate procedure.                                                                                                                                                        |                 |
| <b>Forward Engineering</b> | The process of generating source code from the UML model.                                                                                                                                                                                                                                                               |                 |
| <b>Fork</b>                | Used in State Machine diagrams as pseudo-states.<br><br>With respect to State Machine diagrams, a Fork pseudo-state signifies that its incoming transition comes from a single State, and it has multiple outgoing transitions.                                                                                         | Join            |
| <b>Framework</b>           | A stereotyped package containing model elements that specify a reusable architecture for all or part of a system.                                                                                                                                                                                                       | Pattern         |

| Name | Detail                                                                                                                                                                                          | See also |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|      | <p>Frameworks typically include Classes, Patterns or templates.</p> <p>When frameworks are specialized for an application domain, they are sometimes referred to as Application frameworks.</p> |          |

## 20.7 G

| Name                         | Detail                                                                                                                                                                                                                                                                                                | See also                             |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| <b>Generalizable Element</b> | A model element that can participate in a Generalization relationship.                                                                                                                                                                                                                                | Generalization                       |
| <b>Generalization</b>        | A taxonomic relationship between a more general element and a more specific element. The more specific element is fully consistent with the more general element and contains additional information. An instance of the more specific element can be used where the more general element is allowed. | Generalizable Element<br>Inheritance |
| <b>Guard Condition</b>       | A condition that must be satisfied in order to enable an associated transition to fire.                                                                                                                                                                                                               |                                      |

## 20.8 H

| Name                 | Detail                                                                                                                                                                                                                                                                                                                                     | See also |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>History State</b> | <p>There are two types of History pseudo-states defined in UML: shallow History and deep History.</p> <p>A shallow History sub-state is used to represent the most recently active sub-state of a composite State.</p> <p>A deep History sub-state, in contrast, reflects the most recent active configuration of the composite State.</p> |          |



## 20.9 I

| Name                              | Detail                                                                                                                                                                                                                                                                                                                                                                                        | See also                   |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| <b>Implementation</b>             | A definition of how something is constructed or computed. For example, a Class is an implementation of a type, a method is an implementation of an operation.                                                                                                                                                                                                                                 |                            |
| <b>Implementation Class</b>       | A stereotyped Class that specifies the implementation of a Class in some programming language (for example, C++, Smalltalk, Java) in which an instance can not have more than one Class. An Implementation Class is said to realize a type if it provides all of the operations defined for the type with the same behavior as specified for the type's operations.                           | Type                       |
| <b>Implementation Inheritance</b> | The inheritance of the implementation of a more general element. Includes inheritance of the interface.                                                                                                                                                                                                                                                                                       | Interface Inheritance      |
| <b>Import</b>                     | In the context of packages, a dependency that shows the packages whose Classes can be referenced within a given package (including packages recursively embedded within it).                                                                                                                                                                                                                  | Visibility , Export        |
| <b>Include</b>                    | A relationship from a base Use Case to an inclusion Use Case, specifying how the behavior for the base Use Case contains the behavior of the inclusion Use Case. The behavior is included at the location that is defined in the base Use Case. The base Use Case depends on performing the behavior of the inclusion Use Case, but not on its structure (that is, attributes or operations). | Extend, Expression         |
| <b>Inheritance</b>                | The mechanism by which more specific elements incorporate the structure and behavior of more general elements related by behavior.                                                                                                                                                                                                                                                            | Generalization, Delegation |
| <b>Initial State</b>              | A pseudo-state used to denote the default state of a composite State; there can be one initial vertex in each region of the composite State.                                                                                                                                                                                                                                                  |                            |
| <b>Instance</b>                   | An entity that has a unique identity, a set of operations that can be applied to it, and a state that stores the effects of the operations.                                                                                                                                                                                                                                                   | Object                     |
| <b>Interaction</b>                | A specification of how stimuli are sent between instances to perform a specific task. The interaction is defined in the context of a collaboration.                                                                                                                                                                                                                                           | Collaboration              |

| Name                                | Detail                                                                                                                                                                                                                                                                                                  | See also                   |
|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| <b>Interaction Diagram</b>          | A generic term that applies to several types of diagrams that emphasize object interactions. These include Timing diagrams, Sequence diagrams, Interaction Overview diagrams and Communication diagrams.                                                                                                |                            |
| <b>Interaction Occurrence</b>       | A reference to an existing interaction element. Interaction occurrences are visually represented by a frame, with ref in the frame's title space. The diagram name is indicated in the frame contents.                                                                                                  |                            |
| <b>Interaction Overview Diagram</b> | A diagram that visualizes the cooperation between other Interaction diagrams to illustrate a control flow serving an encompassing purpose. As Interaction Overview diagrams are a variant of Activity diagrams, most of the diagram notation is similar, as is the process in constructing the diagram. |                            |
| <b>Interface</b>                    | A named set of operations that characterize the behavior of an element.                                                                                                                                                                                                                                 | Class , Type               |
| <b>Interface Inheritance</b>        | The inheritance of the interface of a more general element. Does not include inheritance of the implementation.                                                                                                                                                                                         | Implementation Inheritance |
| <b>Internal Transition</b>          | A transition signifying a response to an event without changing the state of an object.                                                                                                                                                                                                                 |                            |
| <b>Interrupt Flow</b>               | An Enterprise Architect-defined toolbox icon used to define the exception handler and interruptible activity region concepts.                                                                                                                                                                           |                            |

## 20.10 J

| Name            | Detail                                                                                                                                                                                                            | See also |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Java</b>     | A fully object-oriented, cross platform language based on elements from Smalltalk, C++ and other OO languages.                                                                                                    |          |
| <b>Join</b>     | Used in State Machine diagrams and in Activity diagrams to synchronize multiple flows.                                                                                                                            | Fork     |
| <b>Junction</b> | Junction pseudo-states are used to design complex transitional paths. A Junction can be used to combine, or merge, multiple paths into a shared transition path or to split an incoming path into multiple paths. |          |

**20.11 L**

| Name              | Detail                                                                                                                                                                                        | See also                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| <b>Layer</b>      | The organization of classifiers or packages at the same level of abstraction. A layer represents a horizontal slice through an architecture, whereas a partition represents a vertical slice. | Partition                      |
| <b>Lifeline</b>   | An individual participant in an interaction (that is, Lifelines cannot have multiplicity). A Lifeline represents a distinct connectable element.                                              |                                |
| <b>Link</b>       | A semantic connector among a tuple of objects.<br>An instance of an Association.                                                                                                              | Association                    |
| <b>Link End</b>   | An instance of an Association end.                                                                                                                                                            | Association End,<br>Classifier |
| <b>Local Path</b> | A relative path on a local machine, enabling developers to store shared source code in machine specific directories, but still generate and synchronize code.                                 |                                |

## 20.12 M

| Name                              | Detail                                                                                                                                                                                                                                       | See also |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Maintenance</b>                | The support of a software system after it is deployed.                                                                                                                                                                                       |          |
| <b>Manifest</b>                   | A relationship that indicates that the artifact source embodies the target model element. Stereotypes can be added to Enterprise Architect to classify the type of manifestation of the model element.                                       |          |
| <b>Message</b>                    | Messages indicate a flow of information, or transition of control, between elements. Messages are used by Communication diagrams, Sequence diagrams, Interaction Overview diagrams and Timing diagrams.                                      |          |
| <b>Message Endpoint</b>           | An element that defines an endpoint of a Lifeline, such as a State or Value Lifeline in a Timing diagram.                                                                                                                                    |          |
| <b>Message Label</b>              | Used for messages sent between Lifelines to make the diagram appear less cluttered. Labels with the same name indicate that a message can be interrupted.                                                                                    |          |
| <b>Metaclass</b>                  | A Class whose instances are Classes. Metaclasses are typically used to construct metamodels.                                                                                                                                                 |          |
| <b>Metafile</b>                   | A vector-based image format native to Windows. Supports high detail and excellent scaling. Typically used for saving diagram images for placement in documents. Comes in Placeable (an older format) and Enhanced (current standard format). |          |
| <b>Meta-Metamodel</b>             | A model that defines the language for expressing a metamodel. The relationship between a meta-metamodel and a metamodel is analogous to the relationship between a metamodel and a model.                                                    |          |
| <b>Metamodel</b>                  | A model that defines the language for expressing a model.                                                                                                                                                                                    |          |
| <b>Meta-Object</b>                | A generic term for all meta-entities in a meta-modeling language. For example, meta-types, meta-classes, meta-attributes, and meta-associations.                                                                                             |          |
| <b>Meta-Object Facility (MOF)</b> | An Object Management Group (OMG) standard. MOF originated in the UML, when the OMG required a Meta-                                                                                                                                          |          |

| Name                         | Detail                                                                                                                                                                                                                                                                                                                  | See also                                                 |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
|                              | Modeling architecture to define the UML. MOF is designed as a four-layered architecture.                                                                                                                                                                                                                                |                                                          |
| <b>Method</b>                | The implementation of an operation. It specifies the algorithm or procedure associated with an operation.                                                                                                                                                                                                               |                                                          |
| <b>Model ( MOF )</b>         | An abstraction of a physical system with a certain purpose.Usage note: In the context of the MOF specification, which describes a meta-metamodel, the meta-metamodel is for brevity frequently referred to simply as the model.                                                                                         | Physical System                                          |
| <b>Model Aspect</b>          | A dimension of modeling that emphasizes particular qualities of the metamodel. For example, the structural model aspect emphasizes the structural qualities of the metamodel.                                                                                                                                           |                                                          |
| <b>Model Elaboration</b>     | The process of generating a repository type from a published model. Includes the generation of interfaces and implementations which enables repositories to be instantiated and populated based on, and in compliance with, the model elaborated.                                                                       |                                                          |
| <b>Model Element ( MOF )</b> | An element that is an abstraction drawn from the system being modeled. In the MOF specification model elements are considered to be meta-objects.                                                                                                                                                                       | View Element                                             |
| <b>Model Library</b>         | A stereotyped package containing model elements that are intended to be reused by other packages. A model library differs from a profile in that a model library does not extend the metamodel using stereotypes and tagged definitions. A model library is analogous to a Class library in some programming languages. |                                                          |
| <b>Modeling Time</b>         | Refers to something that occurs during the modeling phase of the software development process. It includes analysis time and design time.Usage note: When discussing object systems, it is often important to distinguish between modeling-time and run-time concerns.                                                  | Analysis Time<br>Design Time<br>Compile time<br>Run time |
| <b>Module</b>                | A software unit of storage and manipulation. Modules include source code modules, binary code modules and executable code modules.                                                                                                                                                                                      | Component                                                |
| <b>MOF</b>                   | Meta-Object Facility, an Object Management Group (OMG)                                                                                                                                                                                                                                                                  |                                                          |

| Name                           | Detail                                                                                                                                                                                                                                                                                                    | See also                                        |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
|                                | standard. MOF originated in the UML, when the OMG required a Meta-Modeling architecture to define the UML. MOF is designed as a four-layered architecture.                                                                                                                                                |                                                 |
| <b>Multiple Classification</b> | A semantic variation of Generalization in which an object can belong directly to more than one classifier.                                                                                                                                                                                                | Static Classification<br>Dynamic Classification |
| <b>Multiple Inheritance</b>    | A semantic variation of Generalization in which a type can have more than one supertype.                                                                                                                                                                                                                  | Single Inheritance                              |
| <b>Multiplicity</b>            | A specification of the range of enableable cardinalities that a set can assume. Multiplicity specifications can be given for roles within Associations, Parts within Composites, repetitions and other purposes. Essentially a multiplicity is a (possibly infinite) subset of the non-negative integers. | Cardinality                                     |
| <b>Multi-Valued (MOF)</b>      | A model element with multiplicity defined whose Multiplicity Type:: upper attribute is set to a number greater than one. The term multi-valued does not pertain to the number of values held by, for example, an attribute or parameter, at any point in time.                                            | Single-Valued                                   |

## 20.13 N

| Name                     | Detail                                                                                                                                                                                                                                                                   | See also           |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| <b>Name</b>              | A string used to identify a model element.                                                                                                                                                                                                                               |                    |
| <b>Namespace</b>         | A part of the model in which the names can be defined and used. Within a namespace, each name has a unique meaning.                                                                                                                                                      | Name               |
| <b>N-ary Association</b> | An Association among three or more Classes. Each instance of the Association is an n-tuple of values from the respective Classes.                                                                                                                                        | Binary Association |
| <b>Nesting</b>           | A connector used as an alternative membership notation to indicate nested members within an element; for example, a package that has nested members. The nested members of a package could also be shown inside the package rather than linked by the Nesting connector. |                    |
| <b>Node</b>              | A classifier that represents a run-time computation resource, which generally has at least a memory and often processing capability. Run-time objects and components can reside on nodes.                                                                                |                    |



## 20.14 O

| Name                                 | Detail                                                                                                                                                                                                                                                  | See also                                                                                           |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <b>Object</b>                        | An entity with a well-defined boundary and identity that encapsulates state and behavior. State is represented by attributes and relationships, behavior is represented by operations, methods and State Machines. An Object is an instance of a Class. | Class<br>Instance                                                                                  |
| <b>Object Diagram</b>                | A diagram that encompasses objects and their relationships at a point in time. An Object diagram can be considered as a special case of a Class diagram or Communication diagram.                                                                       | Class Diagram<br>Communication Diagram                                                             |
| <b>Object Flow</b>                   | A sub type of the State flow or transition. It implies the passing of an object instance between elements at run-time.                                                                                                                                  |                                                                                                    |
| <b>Object Flow State</b>             | A state in an Activity graph that represents the passing of an object from the output of actions in one State to the input of actions in another State.                                                                                                 |                                                                                                    |
| <b>Object Lifeline</b>               | A line in a Sequence diagram that represents the existence of an object over a period of time.                                                                                                                                                          | Sequence Diagram                                                                                   |
| <b>Object Management Group (OMG)</b> | The standards body responsible for the UML specification and management.                                                                                                                                                                                | Their website is <a href="http://www.omg.org">www.omg.org</a> - follow the links to the UML pages. |
| <b>Occurrence</b>                    | A relationship that indicates that a Collaboration represents a classifier. An Occurrence connector is drawn from the collaboration to the classifier.                                                                                                  |                                                                                                    |
| <b>Operation</b>                     | A service that can be requested from an object to effect behavior. An operation has a signature, which could restrict the actual parameters that are possible.                                                                                          |                                                                                                    |

## 20.15 P

| Name                   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Package</b>         | <p>1. A namespace, as well as an element that can be contained in other packages' namespaces. Packages can own or merge with other packages, and their elements can be imported into a package's namespace.</p> <p>2. A logical container of model elements. It groups elements and can also contain other packages.</p> <p>The <i>OMG UML specification (UML Superstructure Specification, v2.1.1, p. 109)</i> states:</p> <p><i>A package is used to group elements, and provides a namespace for the grouped elements.</i></p> <p><i>A package is a namespace for its members, and can contain other packages. Only packageable elements can be owned members of a package. By virtue of being a namespace, a package can import either individual members of other packages, or all the members of other packages.</i></p> <p><i>In addition a package can be merged with other packages.</i></p> <p>Note that packages own model elements and are the basis for configuration control, storage and access control. Each element can be directly owned by a single package, so the package hierarchy is a strict tree. However, packages can reference other packages, modeled by using one of the stereotypes «import» and «access» of Permission dependency, so the usage network is a graph. Other kinds of dependencies between packages usually imply that one or more dependencies among the elements exist.</p> <p>A package is represented by the common folder icon - a large rectangle with a small rectangle (a 'tab') attached to the left side on top.</p> |          |
| <b>Package Diagram</b> | Used to reflect the organization of packages and their elements, and provide a visualization of their corresponding namespaces.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |          |
| <b>Package Import</b>  | A package import relationship is drawn from a source package to a package whose contents are imported. Private members of a target package cannot be imported.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |          |
| <b>Package Merge</b>   | Indicates a relationship between two packages whereby the contents of the target package are merged with those of the source package. Private contents of a target package are not merged.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |          |

| Name                         | Detail                                                                                                                                                                                                                                                                                                                                                                                                          | See also                                         |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| <b>Parameter</b>             | The specification of a variable that can be changed, passed, or returned. A parameter can include a name, type, and direction. Parameters are used for operations, messages and events.                                                                                                                                                                                                                         | Formal Parameter<br>Argument<br>Actual parameter |
| <b>Parameterized element</b> | The descriptor for a Class with one or more unbound parameters.                                                                                                                                                                                                                                                                                                                                                 | Template<br>Parameterized class                  |
| <b>Parent</b>                | In a generalization relationship, the generalization of another element, the child.                                                                                                                                                                                                                                                                                                                             | Subclass<br>Subtype<br>Child                     |
| <b>Part</b>                  | A run-time instance of a Class or Interface.                                                                                                                                                                                                                                                                                                                                                                    |                                                  |
| <b>Participate</b>           | The connection of a model element to a relationship or to a reified relationship. For example, a Class participates in an Association, an Actor participates in a Use Case.                                                                                                                                                                                                                                     |                                                  |
| <b>Partition</b>             | <ol style="list-style-type: none"> <li>activity graphs: A portion of an activity graph that organizes the responsibilities for actions.</li> <li>architecture: A set of related classifiers or packages at the same level of abstraction or across layers in a layered architecture. A partition represents a vertical slice through an architecture, whereas a layer represents a horizontal slice.</li> </ol> | Layer<br>Swim Lane                               |
| <b>Pattern</b>               | A template collaboration.                                                                                                                                                                                                                                                                                                                                                                                       | Framework                                        |
| <b>Persistent Object</b>     | An object that exists after the process or thread that created it has ceased to exist.                                                                                                                                                                                                                                                                                                                          |                                                  |
| <b>Physical System</b>       | <ol style="list-style-type: none"> <li>The subject of a model.</li> <li>A collection of connected physical units, which can include software, hardware and people, that are organized to accomplish a specific purpose. A physical system can be described by one or more models, possibly from different viewpoints.</li> </ol>                                                                                | Model (MOF)<br>System                            |
| <b>Port</b>                  | Defines the interaction between a classifier and its environment. Interfaces controlling this interaction can be depicted using the 'Expose Interface' toolbox icon.                                                                                                                                                                                                                                            |                                                  |
| <b>Postcondition</b>         | A constraint that must be true at the completion of an                                                                                                                                                                                                                                                                                                                                                          |                                                  |

| Name                   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See also              |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
|                        | operation.                                                                                                                                                                                                                                                                                                                                                                                                                                               |                       |
| <b>Precondition</b>    | A constraint that must be true when an operation is invoked.                                                                                                                                                                                                                                                                                                                                                                                             |                       |
| <b>Primitive Type</b>  | A pre-defined basic datatype without any substructure, such as an integer or a string.                                                                                                                                                                                                                                                                                                                                                                   |                       |
| <b>Process</b>         | <ol style="list-style-type: none"> <li>1. A heavyweight unit of concurrency and execution in an operating system. Thread, which includes heavyweight and lightweight processes. If necessary, an implementation distinction can be made using stereotypes.</li> <li>2. A software development process - the steps and guidelines by which to develop a system.</li> <li>3. To execute an algorithm or otherwise handle something dynamically.</li> </ol> | Thread                |
| <b>Product</b>         | A physical piece of information that is produced by a business or development process. Examples of products include models, source files, scripts, and binary executable files. An product can constitute the implementation of a deployable component.                                                                                                                                                                                                  | Artifact<br>Component |
| <b>Profile</b>         | A stereotyped package that contains model elements that have been customized for a specific domain or purpose using extension mechanisms, such as stereotypes, tagged definitions and constraints. A profile can also specify model libraries on which it depends and the metamodel subset that it extends.                                                                                                                                              |                       |
| <b>Project Browser</b> | The workspace window where the model contents are displayed in 'tree' format. Displays structures such as packages, diagrams and model elements.                                                                                                                                                                                                                                                                                                         |                       |
| <b>Projection</b>      | A mapping from a set to its subset.                                                                                                                                                                                                                                                                                                                                                                                                                      |                       |
| <b>Property</b>        | A named value denoting a characteristic of an element. A property has semantic impact. Certain properties are predefined in the UML; others can be user defined.                                                                                                                                                                                                                                                                                         | Tagged Value          |
| <b>Pseudo-State</b>    | A vertex in a State Machine that has the form of a State, but doesn't behave as a State. Pseudo-states include initial and history vertices.                                                                                                                                                                                                                                                                                                             |                       |

---

| Name                         | Detail                                                                                                                                                                               | See also |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Published Model (MOF)</b> | A model that has been frozen, and that becomes available for instantiating repositories and for support in defining other models. A frozen model's model elements cannot be changed. |          |

**20.16 Q**

| Name             | Detail                                                                                                                                | See also |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Qualifier</b> | An Association attribute or tuple of attributes whose values partition the set of objects related to an object across an Association. |          |

## 20.17 R

| Name                         | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also         |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| <b>Realize</b>               | A source object realizes the destination object.<br><br>Realize is used to express traceability and completeness in the model – a business process or requirement is realized by one or more Use Cases which are in turn realized by some Classes which in turn are realized by a Component, and so on.                                                                                                                                                                             |                  |
| <b>Receive ( A message )</b> | The handling of a stimulus passed from a sender instance.                                                                                                                                                                                                                                                                                                                                                                                                                           | Sender, Receiver |
| <b>Receive</b>               | An element used to define the acceptance or receipt of a request; movement on to the next action occurs until it has received what is defined.                                                                                                                                                                                                                                                                                                                                      |                  |
| <b>Receiver ( Object )</b>   | The object handling a stimulus passed from a sender object.                                                                                                                                                                                                                                                                                                                                                                                                                         | Sender           |
| <b>Reception</b>             | A declaration that a classifier is prepared to react to the receipt of a signal.                                                                                                                                                                                                                                                                                                                                                                                                    |                  |
| <b>Recursion</b>             | A type of message used in Sequence diagrams to indicate a recursive function.                                                                                                                                                                                                                                                                                                                                                                                                       |                  |
| <b>Reference</b>             | 1. A denotation of a model element.<br>2. A named slot within a classifier that facilitates navigation to other classifiers.                                                                                                                                                                                                                                                                                                                                                        | Pointer          |
| <b>Region</b>                | UML 2.x supports both Expansion Regions and Interruptible Activity Regions. <ul style="list-style-type: none"> <li>• An Expansion Region defines the bounds of a region consisting of one or more sets of input collections, where an input collection is a set of elements of the same type</li> <li>• An Interruptible Activity Region contains Activity nodes - when a token leaves an interruptible region, this terminates all of the region's tokens and behaviors</li> </ul> |                  |
| <b>Refinement</b>            | A relationship that represents a fuller specification of something that has already been specified at a certain                                                                                                                                                                                                                                                                                                                                                                     |                  |

| Name                       | Detail                                                                                                                                                                                                                                                                                                                                                                                          | See also                                                                                                                         |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
|                            | level of detail; for example, a design Class is a refinement of an analysis Class.                                                                                                                                                                                                                                                                                                              |                                                                                                                                  |
| <b>Relationship</b>        | A semantic connection among model elements; examples of relationships include Associations and Generalizations.                                                                                                                                                                                                                                                                                 |                                                                                                                                  |
| <b>Repository</b>          | A facility for storing object models, interfaces and implementations.                                                                                                                                                                                                                                                                                                                           |                                                                                                                                  |
| <b>Represents</b>          | A connector that indicates a Collaboration Use is used in a classifier; the connector is drawn from the Collaboration Use to its owning classifier.                                                                                                                                                                                                                                             |                                                                                                                                  |
| <b>Requirement</b>         | A required feature, property or behavior of a system (external requirement).                                                                                                                                                                                                                                                                                                                    |                                                                                                                                  |
| <b>Responsibility</b>      | A contract or obligation of a classifier (internal requirement).                                                                                                                                                                                                                                                                                                                                |                                                                                                                                  |
| <b>Reuse</b>               | The use of a pre-existing artifact.                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                  |
| <b>Reverse Engineering</b> | The process of importing source code into the model as standard UML model objects (such as Classes, attributes and operations).                                                                                                                                                                                                                                                                 |                                                                                                                                  |
| <b>Rich Text Format</b>    | A standard mark-up language for creating word processor documents, frequently associated with Microsoft Word.                                                                                                                                                                                                                                                                                   |                                                                                                                                  |
| <b>Robustness Diagram</b>  | Enterprise Architect supports business process modeling extensions from the UML business process model profile; Robustness diagrams are used in ICONIX.                                                                                                                                                                                                                                         | Read more about this at <a href="http://www.sparxsystems.com/iconix/iconixsw.htm">www.sparxsystems.com/iconix/iconixsw.htm</a> . |
| <b>Role</b>                | <ol style="list-style-type: none"> <li>1. The named detail and rules associated with one end of an association; a role can indicate name, constraints, multiplicity and collection details.</li> <li>2. The named specific behavior of an entity participating in a particular context; a role can be static (such as an Association end) or dynamic (such as a Collaboration role).</li> </ol> |                                                                                                                                  |
| <b>Role Binding</b>        | The mapping between a Collaboration Use's internal                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                  |



---

| Name            | Detail                                                                                                                                                                                           | See also                                                |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
|                 | roles and the respective parts required to implement a specific situation; the associated parts can have properties defined to enable the binding to occur, and the collaboration to take place. |                                                         |
| <b>Run Time</b> | The period of time during which a computer program executes.                                                                                                                                     | Analysis Time, Compile Time, Design Time, Modeling Time |

## 20.18 S

| Name                            | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See also                               |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| <b>Scenario</b>                 | <ol style="list-style-type: none"> <li>1. A specific sequence of actions that illustrates behaviors. A scenario can be used to illustrate an interaction or the execution of a Use Case instance.</li> <li>2. A sequence of operations carried out in some order to produce a known result. Can apply to Use Cases where it is the equivalent of a Sequence diagram, or to other objects to describe how they are used at run-time.</li> </ol>                                                                                                                                                  | Interaction                            |
| <b>Schema ( MOF )</b>           | In the context of the MOF, analogous to a package that is a container of model elements. Schema corresponds to a MOF package.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Metamodel, Package                     |
| <b>Self-Message</b>             | Reflects a new process or method invoked within the calling Lifeline's operation. It is a specification of a message.                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                        |
| <b>Semantic Variation Point</b> | A point of variation in the semantics of a metamodel. It provides an intentional degree of freedom for the interpretation of the metamodel semantics.                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                        |
| <b>Send (A message )</b>        | The passing of a stimulus from a sender instance to a receiver instance.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Sender, Receiver                       |
| <b>Sender ( Object )</b>        | The object passing a stimulus to a receiver object.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Receiver                               |
| <b>Sequence Diagram</b>         | <p>A diagram that shows object interactions arranged in time sequence. In particular, it shows the objects participating in the interaction and the sequence of messages exchanged.</p> <p>Unlike a Communication (Collaboration) diagram, a Sequence diagram includes time sequences but does not include object relationships. A Sequence diagram can exist in a generic form (describes all possible scenarios) and in an instance form (describes one actual scenario).</p> <p>Sequence diagrams and Communication diagrams express similar information, but show it in different ways.</p> | Communication Diagram, Object Lifeline |
| <b>Signal</b>                   | The specification of an asynchronous stimulus communicated between instances. Signals can have parameters.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                        |
| <b>Signature</b>                | The name and parameters of a behavioral feature. A signature can include an optional returned parameter.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                        |

| Name                         | Detail                                                                                                                                                                                                                                                                                                                                                                        | See also                       |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
|                              |                                                                                                                                                                                                                                                                                                                                                                               |                                |
| <b>Single Inheritance</b>    | A semantic variation of Generalization in which a type can have only one supertype.                                                                                                                                                                                                                                                                                           | Multiple Inheritance           |
| <b>Single Valued ( MOF )</b> | A model element with multiplicity defined is single valued when its Multiplicity Type: upper attribute is set to 1.<br><br>The term single-valued does not pertain to the number of values held by, for example, an attribute or parameter at any point in time, since a single-valued attribute (for instance, with a multiplicity lower bound of zero) could have no value. | Multi-Valued                   |
| <b>Specification</b>         | A declarative description of what something is or does.                                                                                                                                                                                                                                                                                                                       | Implementation                 |
| <b>State</b>                 | A condition or situation during the life of an object during which it satisfies some condition, performs some activity, or waits for some event.                                                                                                                                                                                                                              | State ( OMA )                  |
| <b>State Invariant</b>       | A condition applied to a Lifeline that must be fulfilled for the Lifeline to exist.                                                                                                                                                                                                                                                                                           |                                |
| <b>State Machine</b>         | A behavior that specifies the sequences of States that an object or an interaction goes through during its life in response to events, together with its responses and actions.                                                                                                                                                                                               |                                |
| <b>State Machine Diagram</b> | A diagram that illustrates how an element, often a Class, can move between States classifying its behavior, according to transition triggers, constraining guards and other aspects of State Machine diagrams that depict and explain movement and behavior.                                                                                                                  |                                |
| <b>State Chart</b>           | A diagram that shows a State Machine.                                                                                                                                                                                                                                                                                                                                         | State Machine , Activity Graph |
| <b>State Continuation</b>    | A symbol that serves two different purposes for Interaction diagrams - as State Invariants and as Continuations.<br><br>A State Invariant is a condition applied to a Lifeline that must be fulfilled for the Lifeline to exist.<br><br>A Continuation is used in seq and alt combined fragments to indicate the branches of continuation that an operand follows.            |                                |
| <b>State Lifeline</b>        | A State Lifeline follows discrete transitions between                                                                                                                                                                                                                                                                                                                         |                                |

| Name                           | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                       | See also                                            |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
|                                | States, which are defined along the y-axis of the timeline. Any transition has optional attributes of timing constraints, duration constraints and observations.                                                                                                                                                                                                                                                                             |                                                     |
| <b>Static Classification</b>   | A semantic variation of Generalization in which an object can not change classifier.                                                                                                                                                                                                                                                                                                                                                         | Multiple Classification ,<br>Dynamic Classification |
| <b>Stereotype</b>              | <p>A new type of modeling element that extends the semantics of the metamodel.</p> <p>Stereotypes must be based on certain existing types or Classes in the metamodel.</p> <p>Stereotypes can extend the semantics, but not the structure of pre-existing types and Classes. Certain stereotypes are predefined in the UML, others can be user defined. Stereotypes are one of three extensibility mechanisms in UML.</p>                    | Constraint , Tagged value                           |
| <b>Stimulus</b>                | The passing of information from one instance to another, such as raising a signal or invoking an operation. The receipt of a signal is normally considered an event.                                                                                                                                                                                                                                                                         | Message                                             |
| <b>String</b>                  | A sequence of text characters. The details of string representation depend on implementation, and can include character sets that support international characters and graphics.                                                                                                                                                                                                                                                             |                                                     |
| <b>Structural Diagram</b>      | <p>A diagram that depicts the structural elements composing a system or function.</p> <p>These diagrams can reflect the static relationships of a structure, as do Class or Package diagrams, or run-time architectures, such as Object or Composite Structure diagrams.</p> <p>Structural diagrams include Class diagrams, Composite Structure diagrams, Component diagrams, Deployment diagrams, Object diagrams and Package diagrams.</p> |                                                     |
| <b>Structural Feature</b>      | A static feature of a model element, such as an attribute.                                                                                                                                                                                                                                                                                                                                                                                   |                                                     |
| <b>Structural Model Aspect</b> | A model aspect that emphasizes the structure of the objects in a system, including their types, Classes, relationships, attributes and operations.                                                                                                                                                                                                                                                                                           |                                                     |
| <b>Subactivity State</b>       | A State in an activity graph that represents the execution of a non-atomic sequence of steps that has some duration.                                                                                                                                                                                                                                                                                                                         |                                                     |

| Name                    | Detail                                                                                                                                                                                                                                                       | See also                                                  |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| <b>Subclass</b>         | In a Generalization relationship, the specialization of another Class; the superclass.                                                                                                                                                                       | Generalization, Child , Parent Superclass                 |
| <b>Submachine State</b> | A State in a State Machine that is equivalent to a composite State but its contents are described by another State Machine.                                                                                                                                  |                                                           |
| <b>Subpackage</b>       | A package that is contained in another package.                                                                                                                                                                                                              |                                                           |
| <b>Substate</b>         | A State that is part of a composite State.                                                                                                                                                                                                                   | Composite State , Concurrent Substate , Disjoint Substate |
| <b>Subsystem</b>        | A grouping of model elements that represents a behavioral unit in a physical system.<br><br>A subsystem offers interfaces and has operations. In addition, the model elements of a subsystem can be partitioned into specification and realization elements. | Package, Physical System                                  |
| <b>Subtype</b>          | In a Generalization relationship, the specialization of another type; the supertype.                                                                                                                                                                         | Generalization , Child , Parent Supertype                 |
| <b>Superclass</b>       | In a Generalization relationship, the generalization of another Class; the subclass.                                                                                                                                                                         | Generalization Subclass                                   |
| <b>Supertype</b>        | In a Generalization relationship, the generalization of another type; the subtype.                                                                                                                                                                           | Generalization Subtype                                    |
| <b>Supplier</b>         | A classifier that provides services that can be invoked by others.                                                                                                                                                                                           | Client                                                    |
| <b>Swimlane</b>         | A partition on an Activity diagram for organizing the responsibilities for actions. Swimlanes typically correspond to organizational units in a business model.                                                                                              | Partition                                                 |
| <b>Synch</b>            | A State used for indicating that concurrent paths of a State Machine are synchronized. After bringing the paths to a synch state, the emerging transition indicates unison.                                                                                  |                                                           |
| <b>Synchronize Code</b> | The process of importing and exporting code changes to ensure the model and source code match.                                                                                                                                                               |                                                           |

| Name                   | Detail                                                        | See also        |
|------------------------|---------------------------------------------------------------|-----------------|
| <b>System</b>          | A top-level subsystem in a model.                             | Physical System |
| <b>System Boundary</b> | An element used to delineate a particular part of the system. |                 |

## 20.19 T

| Name                         | Detail                                                                                                                                                                                                                                                                                                 | See also                                   |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| <b>Table</b>                 | A relational table (composed of columns).                                                                                                                                                                                                                                                              |                                            |
| <b>Tagged Value</b>          | <p>The explicit definition of a property as a name-value pair.</p> <p>In a Tagged Value, the name is referred to as the tag. Certain tags are predefined in the UML; others can be user defined.</p> <p>Tagged Values are one of three extensibility mechanisms in UML.</p>                            | Constraint, Property, Stereotype           |
| <b>Template</b>              | The descriptor for a Class with one or more unbound parameters.                                                                                                                                                                                                                                        | Parameterized Element, Parameterized Class |
| <b>Terminate</b>             | A pseudostate indicating that upon entry of its pseudostate, the State Machine's execution ends.                                                                                                                                                                                                       |                                            |
| <b>Thread ( of Control )</b> | A single path of execution through a program, a dynamic model, or some other representation of control flow. Also, a stereotype for the implementation of an active object as a lightweight process.                                                                                                   | Active Object, Process, Concurrency        |
| <b>Time Event</b>            | An event that denotes the time elapsed since the current state was entered.                                                                                                                                                                                                                            | Event                                      |
| <b>Time Expression</b>       | An expression that resolves to an absolute or relative value of time.                                                                                                                                                                                                                                  |                                            |
| <b>Timing Diagram</b>        | A diagram that defines the behavior of different objects within a time-scale, with visual depictions of those objects changing state and interacting over time.                                                                                                                                        |                                            |
| <b>Toolbox</b>               | The main toolbar running down the center of Enterprise Architect, from which you can select model elements to insert into diagrams. This is also known as the Diagram Toolbox and the Object toolbar.                                                                                                  |                                            |
| <b>Top Level</b>             | <p>A stereotype of package denoting the top-most package in a containment hierarchy.</p> <p>The topLevel stereotype defines the outer limit for looking up names, as namespaces 'see' outwards. For example, <b>opTopLevelSubsystem</b> represents the top of the subsystem containment hierarchy.</p> |                                            |

| Name                    | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                               | See also                        |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| <b>Trace</b>            | A dependency that indicates a historical or process relationship between two elements that represent the same concept without specific rules for deriving one from the other.                                                                                                                                                                                                                                                                        |                                 |
| <b>Transient Object</b> | An object that exists only during the execution of the process or thread that created it.                                                                                                                                                                                                                                                                                                                                                            |                                 |
| <b>Transition</b>       | A relationship between two States indicating that an object in the first State performs certain specified actions and enters the second State when a specified event occurs and specified conditions are satisfied. On such a change of State, the transition is said to fire.                                                                                                                                                                       | Fire, Object Flow               |
| <b>Type</b>             | <p>A stereotyped Class that specifies a domain of objects together with the operations applicable to the objects, without defining the physical implementation of those objects.</p> <p>A type can not contain any methods, maintain its own thread of control, or be nested. However, it can have attributes and associations.</p> <p>Although an object can have at most one implementation Class, it can conform to multiple different types.</p> | Implementation Class, Interface |
| <b>Type Expression</b>  | An expression that evaluates to a reference to one or more types.                                                                                                                                                                                                                                                                                                                                                                                    |                                 |



## 20.20 U

| Name                      | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See also          |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <b>UML</b>                | <p>The Unified Modeling Language, a notation and specification for modeling software systems in an Object-Oriented manner.</p> <p>You can read more about UML on the OMG home page or in our UML Tutorial.</p>                                                                                                                                                                                                                                                                                  |                   |
| <b>UML Diagrams</b>       | <p>Diagrams used to model different aspects of the system under development. They include various elements and connectors, all of which have their own meanings and purposes.</p> <p>UML 2.3 includes 14 diagrams: Use Case diagram, Activity diagram, State Machine diagram, Timing diagram, Sequence diagram, Interaction Overview diagram, Communication diagram, Package diagram, Class diagram, Object diagram, Composite Structure diagram, Component diagram and Deployment diagram.</p> |                   |
| <b>UML Toolbox</b>        | <p>The main toolbar running down the center of Enterprise Architect from which you can select model elements to insert into diagrams. This is also known as the Diagram Toolbox and the Object toolbar.</p>                                                                                                                                                                                                                                                                                     |                   |
| <b>Uninterpreted</b>      | <p>A placeholder for a type or types whose implementation is not specified by the UML. Every uninterpreted value has a corresponding string representation.</p>                                                                                                                                                                                                                                                                                                                                 | Any ( CORBA )     |
| <b>Usage</b>              | <p>A dependency in which one element (the client) requires the presence of another element (the supplier) for its correct functioning or implementation.</p>                                                                                                                                                                                                                                                                                                                                    |                   |
| <b>Use</b>                | <p>A connector that indicates that one element requires another to perform some interaction. The Usage relationship does not specify how the target supplier is used, other than that the source client uses it in definition or implementation.</p>                                                                                                                                                                                                                                            |                   |
| <b>Use Case ( Class )</b> | <p>A UML model element that describes how a user of the proposed system interacts with the system to perform a discrete unit of work. It describes and signifies a single interaction over time that has meaning for the end user (person, machine or other system), and is required to leave the system in a complete state: either the interaction completed or was rolled back to the initial state.</p>                                                                                     | Use Case Instance |
| <b>Use Case</b>           | <p>A diagram that captures Use Cases and Actor</p>                                                                                                                                                                                                                                                                                                                                                                                                                                              |                   |

| Name                       | Detail                                                                                                                                                                                                                                                                                          | See also       |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| <b>Diagram</b>             | interactions. It describes the functional requirements of the system, the manner in which outside things (Actors) interact at the system boundary, and the response of the system.                                                                                                              |                |
| <b>Use Case Estimation</b> | The technique of estimating project size and complexity based on the number of Use Cases and their difficulty.                                                                                                                                                                                  |                |
| <b>Use Case Instance</b>   | The performance of a sequence of actions being specified in a Use Case. An instance of a Use Case.                                                                                                                                                                                              | Use Case Class |
| <b>Use Case Model</b>      | A model that describes a system's functional requirements in terms of Use Cases.                                                                                                                                                                                                                |                |
| <b>Utility</b>             | <p>A stereotype that groups global variables and procedures in the form of a Class declaration. The utility attributes and operations become global variables and global procedures, respectively.</p> <p>A utility is not a fundamental modeling construct, but a programming convenience.</p> |                |

## 20.21 V

| Name                   | Detail                                                                                                                                                                                  | See also            |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <b>Value</b>           | An element of a type domain.                                                                                                                                                            |                     |
| <b>Value Lifeline</b>  | A Lifeline that shows the Lifeline's state across the diagram, within parallel lines indicating a steady state.<br>A cross between the lines indicates a transition or change in state. |                     |
| <b>Vertex</b>          | A source or a target for a transition in a State Machine; a vertex can be either a State or a pseudo-state.                                                                             | State, Pseudo-State |
| <b>View</b>            | A projection of a model, which is seen from a given perspective or vantage point and omits entities that are not relevant to this perspective.                                          |                     |
| <b>View Element</b>    | An element that is a textual and/or graphical projection of a collection of model elements.                                                                                             | Model Element (MOF) |
| <b>View Projection</b> | A projection of model elements onto view elements.<br>A View Projection provides a location and a style for each view element.                                                          |                     |
| <b>Visibility</b>      | An enumeration whose value (public, protected, package or private) denotes how the model element to which it refers can be seen outside its enclosing namespace.                        | Export, Import      |
| <b>Visual Basic</b>    | A rapid application development programming language.<br>Windows' only scripting language based on COM.                                                                                 |                     |

**Part**



## 21 License Management



The License Management dialog in Enterprise Architect enables you to upgrade Enterprise Architect and to register Add-Ins. The dialog lists:

- The currently-registered keys
- The product each key applies to
- When each private key expires
- When each shared key is to be reactivated in the keystore for issue to another user

**Access** [Help | Register and Manage License Key\(s\)](#)


### Reference

The options on the License Management dialog are explained in the following table:

| Option                                  | Action                                                                                                                                                                                                                                                                                                                                                                           | See also                                                                                                                                                                                                                            |
|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Add Key</b>                          | Display the Add Registration Key dialog, which enables you to: <ul style="list-style-type: none"> <li>• Register your new Enterprise Architect license</li> <li>• Obtain a shared key from the Sparx Systems Key Store (available for version 4.51 and above)</li> <li>• Add a new key to update to a higher version of Enterprise Architect or to register an Add-In</li> </ul> | <a href="#">Register a Full License</a> <sup>[38]</sup><br><a href="#">Add License Keys</a> <sup>[2739]</sup><br><a href="#">Upgrade an Existing License</a> <sup>[2743]</sup><br><a href="#">Register Add-In</a> <sup>[2745]</sup> |
| <b>Remove Key</b><br><b>Release Key</b> | <ul style="list-style-type: none"> <li>• (Private Key) Make the Add-In or current version of Enterprise Architect inoperable</li> <li>• (Shared Key) Release the key to the keystore; however, the Keystore Manager is normally configured to release keys automatically when the user logs off</li> </ul>                                                                       |                                                                                                                                                                                                                                     |
| <b>Copy</b>                             | Place the highlighted key into the clipboard.                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                     |
| <b>Close</b>                            | Close the dialog.                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                     |
| <b>Help</b>                             | Display the help for this topic.                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                     |

| Option | Action | See also |
|--------|--------|----------|
|        |        |          |

**Learn more**

- [Finding Your License Information](#)  <sup>[2738]</sup>
- [Keystore Troubleshooting](#)  <sup>[2742]</sup>

## 21.1 Finding Your License Information

You can find information on your Enterprise Architect license in the About Enterprise Architect dialog.

[Access](#) [Help](#) | [About EA](#)

## 21.2 Adding License Keys

Two types of license key can be used in conjunction with Enterprise Architect: Private and Shared.

### Reference

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also                                                      |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| <b>Private Keys</b>  | <p>Private keys enable you to register, indefinitely, on the machine and user account that you are currently using:</p> <ul style="list-style-type: none"> <li>• An Enterprise Architect license for the edition of Enterprise Architect you have purchased, or</li> <li>• An Add-In product license (such as for MDG Link for Eclipse or MDG Link for Visual Studio.NET).</li> </ul>                                                                                                                                                                                                                                                                                                                                                          | <a href="#">Enterprise Architect Editions</a> <sup>[20]</sup> |
| <b>Shared Keys</b>   | <p>Shared keys are obtained from a central shared key store, to give you temporary access to the installed 'suite' edition of Enterprise Architect.</p> <p>Shared Keys are available with the purchase of a floating license, and require Enterprise Architect version 4.51 or higher.</p> <p>Shared Keys require a shared license key store to be configured by your license administrator; the key store can be either file-based or network based (preferred).</p> <p>Only the Key Administrator is required to install the Sparx Enterprise Key Store application; end users simply connect to the configured key file - advised by the administrator - as described below.</p> <p>No additional software is required to be installed.</p> | <a href="#">Enterprise Architect Floating License.</a>        |
| <b>Notes on Keys</b> | <p>Some license keys can override and remove others</p> <ul style="list-style-type: none"> <li>• The key for a more advanced edition of Enterprise Architect replaces the key for a simpler edition</li> <li>• The key for MDG Integration for Visual Studio replaces the key for MDG Link for Visual Studio</li> </ul> <p>Shared keys and Private keys have different formats and cannot be used interchangeably.</p>                                                                                                                                                                                                                                                                                                                         |                                                               |

**Access** [Help | Register and Manage License Key\(s\) > License Management: Add Key > Add Registration Key](#)

**How to**



To add a private key

| Step | Action                                                                                                                               | See also                                   |
|------|--------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| 1    | Click on the Enter Private Key tab.                                                                                                  | <a href="#">License Management</a><br>2736 |
| 2    | In the <b>Name</b> and <b>Company</b> fields, type your user name and company name.                                                  |                                            |
| 3    | In the <b>Copy registration key into space below...</b> field, copy the registration key (including any parentheses around the key). |                                            |
| 4    | Click on the <b>OK</b> button to confirm the key selection.                                                                          |                                            |

To add a shared key

| Step | Action                                                                                                                                                                                                                                                | See also                                   |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| 1    | Click on the Get Shared Key tab.                                                                                                                                                                                                                      | <a href="#">License Management</a><br>2736 |
| 2    | In the <b>Name</b> and <b>Company</b> fields, type your user name and company name.                                                                                                                                                                   |                                            |
| 3    | In the <b>Shared Keystore</b> field, click on the ( ... ) (Browse) button.<br>The Shared Keystore Selection dialog displays.                                                                                                                          |                                            |
| 4    | If your keystore is file-based: <ul style="list-style-type: none"> <li>• Select the <b>File Based Keystore</b> radio button</li> <li>• Click on the <b>Browse</b> button, and</li> <li>• Locate and select the keystore file</li> </ul> Go to step 7. |                                            |
| 5    | If the keystore is network-based: <ul style="list-style-type: none"> <li>• Click on the <b>Sparx Keystore Server</b> radio button</li> <li>• In the <b>Server Address</b> field, type the server address of the keystore.</li> </ul>                  |                                            |
| 6    | If necessary, type in the password (advised by your administrator) and/or click on the <b>Test</b> button to ensure that you have a connection to the                                                                                                 |                                            |

| Step | Action                                                                                                                                                                      | See also |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|      | keystore.                                                                                                                                                                   |          |
| 7    | Click on the <b>OK</b> button to return to the Get Shared Key tab.<br>This now shows the name of the keystore in the <b>Shared Keystore</b> field.                          |          |
| 8    | In the <b>Select a Product</b> field, click on the appropriate product name.                                                                                                |          |
| 9    | Click on the <b>OK</b> button.<br>The License Management dialog redisplay, indicating that the shared key is registered for the selected product until the key expiry date. |          |
| 10   | Click on the <b>Close</b> button.                                                                                                                                           |          |

## 21.3 Keystore Troubleshooting

| Message Displayed:                                                        | Explanation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><i>Error reading Key Store file:<br/>(Access is denied)</i></p>        | <p>All users who are to use the shared key facility require <b>Read, Write and Modify</b> access to the <i>sskeys.dat</i> file containing the shared keys.</p> <p>Please verify that all required users have sufficient permissions to the file and try again.</p> <p>Review the effective permissions calculated at the location of the key file for the user account reporting the problem - you should closely examine the permissions for both the Network Share and the File System; it is possible that these permissions have been overwritten at some point.</p> <p>If the problem continues, contact Sparx Support.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <p><i>Error reading Key Store file:<br/>(Key File has been moved)</i></p> | <p>In the key store, as a security measure the hard drive serial number is recorded when the file is created; the file then cannot be moved from the original location in which it was created.</p> <p>If the key store has to be re-located for any reason, the administrator should re-create the key store in the new location using the original license keys.</p> <p>This issue is commonly seen after a file server has undergone a hardware upgrade in which the physical hard drives have been replaced; problems could also occur if the drive used is part of a RAID configuration.</p> <p>The message can also appear where the key store exists on a Novell-based file system - when creating the key store, the administrator is prompted to confirm that the key store is to be located on a Novell Netware file server; if they click on the <b>Yes</b> button, the key store instead records the logical path used to create it, and all users must connect to the key store using this same path.</p> <p>The recorded path is case-sensitive and must be an exact match.</p> |

## 21.4 Upgrade an Existing License

This topic explains how to upgrade your installation of Enterprise Architect to a more powerful edition.

| Topic                          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                             | See also                                                                                                                                                                    |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Editions</b>                | <p>There are six Enterprise Architect editions:</p> <ul style="list-style-type: none"> <li>• Desktop</li> <li>• Professional</li> <li>• Corporate</li> <li>• Business and Software Engineering</li> <li>• Systems Engineering</li> <li>• Ultimate</li> </ul> <p>The installation file provides the facilities of all editions; your registration key gives access to the facilities of the edition you have purchased.</p>                         | <a href="#">Editions Available</a> <sup>[22]</sup>                                                                                                                          |
| <b>Upgrade</b>                 | <p>If you have purchased a less powerful edition, such as Professional or Desktop, you can subsequently upgrade your existing license to a more powerful edition by purchasing a special upgrade key from Sparx Systems.</p> <p>Once you have purchased and received the appropriate key, use the procedure below to gain access to the additional features.</p>                                                                                   | <a href="#">Enterprise Architect Purchase Details</a>                                                                                                                       |
| <b>Trial and Lite Editions</b> | <p>The Lite edition and the Trial version cannot be registered or upgraded.</p> <p>If you have used either of these versions and then purchased Enterprise Architect, you must download the registered version from:</p> <p style="text-align: center;"><i><a href="http://www.sparxsystems.com/securedownloads/easetupfull.exe">www.sparxsystems.com/securedownloads/easetupfull.exe</a></i></p> <p>Then you can enter your registration key.</p> | <a href="#">The Trial Version</a> <sup>[20]</sup><br><a href="#">The Read-only 'Lite' Edition</a> <sup>[26]</sup><br><a href="#">Registered User software download site</a> |
| <b>Application</b>             | <p>This topic is mainly applicable to users with private keys.</p> <p>If you are an end-user with a shared key, you would simply be allocated the relevant key next time you requested one.</p> <p>If, however, you want to upgrade while using a shared key on a long lease, you would simply click on the <b>Release Key</b> button and then the <b>Add Key</b> button.</p>                                                                      | <a href="#">Adding License Keys</a> <sup>[2738]</sup>                                                                                                                       |

Access [Help | Register and Manage License Key\(s\)](#) > **License Management: Add Key**

How to

To upgrade from one license edition to another

| Step | Action                                                                                                                                                                                                                                                                                                                                             | See also                                   |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| 1    | Make sure you have a valid upgrade key purchased from Sparx Systems; you typically receive this in an email or PDF format.                                                                                                                                                                                                                         |                                            |
| 2    | On the Add Registration Key dialog, in the <b>Name</b> and <b>Company</b> fields, type your name and company name.                                                                                                                                                                                                                                 | <a href="#">License Management</a><br>2736 |
| 3    | Copy the key you received for the upgraded edition of Enterprise Architect - including the { and } bracket characters - from the email.                                                                                                                                                                                                            |                                            |
| 4    | Paste the key into the <b>Copy registration key</b> field.                                                                                                                                                                                                                                                                                         |                                            |
| 5    | Click on the <b>OK</b> button.<br><br>Enterprise Architect displays the message:<br><br><i>Registration succeeded – Thank you for purchasing Enterprise Architect<br/>xxxx Edition</i>                                                                                                                                                             |                                            |
| 6    | Click on the <b>OK</b> button, and then on the <b>Close</b> button to continue working in Enterprise Architect.                                                                                                                                                                                                                                    |                                            |
| 7    | Select the <b>Help   About EA</b> menu option.<br><br>Copy the registration key shown and store it somewhere safe; this is a key to the full license of the edition you have upgraded to.<br><br>If you ever have to reinstall Enterprise Architect, you can register it with this key, so you won't have to go through the upgrade process again. |                                            |

## 21.5 Register Add-In

Apart from registering Enterprise Architect, you must also register any Add-Ins you purchase to use with the application

This topic explains how to register a private key; if you have purchased a floating licence (shared key) for the Add-In, follow the *Add a Shared Key* section of the *Adding License Keys* topic.



**Access** [Help | Register and Manage License Key\(s\) Extensions | Enter License Key for <Add-In name>](#)

### How to

To register Add-Ins for Enterprise Architect

| Step | Action                                                                                                                                                                                                                                                                     | See also                                                                |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| 1    | Purchase one or more licenses for the Add-In from your Add-In provider.<br>Once you have paid for a licensed version of the Add-In, you receive (via email or other suitable means) a license key for the product.                                                         |                                                                         |
| 2    | Save the license key and the latest full version of the Add-In.                                                                                                                                                                                                            |                                                                         |
| 3    | Run the Add-In's setup program to install the Add-In.                                                                                                                                                                                                                      |                                                                         |
| 4    | In Enterprise Architect, select either of the menu options: <ul style="list-style-type: none"> <li>• <b>Register and Manage License Key(s)</b></li> <li>• <b>Extensions   Enter License Key for &lt;Add-In name&gt;</b></li> </ul> The License Management dialog displays. | <a href="#">License Management</a><br><small>t<sup>[2736]</sup></small> |
| 5    | Click on the <b>Add Key</b> button.<br>The Add Registration Key dialog displays, showing the Enter Private Key tab.                                                                                                                                                        |                                                                         |
| 6    | Type or copy-and-paste in the key you received with the Add-In, including the { and } characters.                                                                                                                                                                          |                                                                         |
| 7    | Click on the <b>OK</b> button.                                                                                                                                                                                                                                             |                                                                         |
| 8    | When the Add-In has been added successfully, close down Enterprise Architect and restart it to apply the integration changes imposed by the Add-In.                                                                                                                        |                                                                         |

**Learn more**

- [Adding License Keys](#) 
- [Add-In License Management Events](#) 

# Index

■ ■ ■

## .EMX

Import 417

## .NET

ASP, Debug 1853  
 Debug 1850  
 Debug Another Process 1888  
 Debug CLR Versions 1851  
 Debug With COM Interop Process 1852  
 Debug, System Requirements 1837  
 Garbage Collect (Exit Method) 2448  
 Set Up Debug Session 1850

## .UML2

Import 417

## - A -

### Abstract

Complex Models 1194  
 XSD Models 2046

### Acceptance Testing 2220

### AcceptEvent Action

Triggers Tab 1095

### Access

Email 165  
 Internet Search Engine 165  
 MDG Technologies Remote From Enterprise Architect 1287  
 Web Site 165

### Access 2007

Repository, Create 194  
 Upsize To 227

### Acknowledgement

CXImage Library 34  
 Of Contributions 34  
 Of Trademarks 31  
 Print Listview 34

### Action

AcceptEvent, Triggers Tab 1095  
 As Hyperlink 1670  
 BroadcastSignal, Signal Tab 1095  
 Call 1787  
 Element 1091  
 Expansion Node 1099  
 Invocation 1787  
 Local Pre/Post Conditions 1093

Notation 1095  
 Operations 1091, 1100  
 SendSignal, Signal Tab 1095  
 StructuralFeature 1095  
 Trigger, AcceptEvent 1095  
 Type, Set 1095  
 Update Operation 1091, 1100

### Action Pin

Add To Action 1097  
 As Action Property 1098  
 As Argument For Call Action 1097  
 Assign To Action 1098  
 Properties 1097

### ActionScript

Code Generation Language Options 1909  
 Import, Reverse Engineering 1793  
 Modeling Conventions 1742  
 Options 1909  
 Versions Supported 1909

### Activate

MDG Technologies 1285

### Activate Recording Markers 1869

### Activation

End 1080  
 Extend Down 1080  
 Extend Up 1080  
 Lower 1080  
 Raise 1080  
 Sequence Element 1080  
 Suppress 1080

### Activation Layer

Sequence Diagram Lifelines 1082

### Activation Levels

Sequence Diagram Lifeline Self Messages 1082

### Active

Classes 1182  
 State Configuration 1153

### Active Directory

Import User ID From 257

### Active State Logic

Model State Machine For HDL 1784

### Activity

Behavioral Aspects 910  
 Edit Parameters 910  
 Element 1101  
 Elements And Connectors 712  
 Group, Toolbox 712  
 Instance 1260  
 Notation 1103  
 Parameter Node 1103  
 Parameters, Define 913



- Activity
  - Partition 1105, 1148
  - Paste As Action 736
  - Paste As Link 736
  - Paste From Project Browser 736
  - Process Element 1674
  - Region Element 1150
  - Structured 1163
  - Structured, Conditional Node 1161, 1168
  - Structured, Loop Node 1161, 1164
  - Structured, Sequential 1161
- Activity (BPEL 1.1)
  - Create 1581
  - Loop 1581
  - Model 1581
  - Sub-Process 1581
  - Task 1581
  - Types 1581
- Activity (BPEL 2.0)
  - Create 1605
  - Loop 1605
  - Model 1605
  - Sub-Process 1605
  - Task 1605
  - Types 1605
- Activity Diagram
  - Create Object From Attribute 896
  - Description 1028
  - Elements And Connectors 1028
  - Example 1029
  - Generate Code From 1778, 1787
  - Generate From Scenario 864, 865
  - Generate Scenario From 873
  - Object Flows 1250
  - Operations 1091, 1100
- Activity Edge
  - Connector 1230
  - Relationship 1230
- Activity Final
  - Element 1127
- Activity Parameter Node
  - Exception 1103
  - Stream 1103
- Activity Partition
  - Docking 1148
  - Element 1148
  - Horizontal 1148
  - Vertical 1148
- Actor
  - Element 1107
- Ada 2005
  - Code Generation Language Options 1909
- Modeling Conventions 1743
- Options 1909
- Adaptive Server Anywhere
  - Data Repository, Connect To 219
  - ODBC Driver, Set Up 205
  - Repository, Create 198
  - Upsize To 234
- Add
  - Category To Team Review 284
  - Code Modules In MDG Technology Wizard 1331
  - Connectors Between Locked Elements 273
  - Connectors To UML Model, Quick Start 55
  - Diagram Properties Note 747
  - Diagram To Package 52
  - Diagram to Project 724
  - Diagram To UML Model, Quick Start 52
  - Diagram Type In MDG Technology Wizard 1329
  - Element Changes 2238
  - Element Defects 2238
  - Element Directly To Package 798
  - Element Issues 2238
  - Element Tasks 2238
  - Element To Diagram 53
  - Element To Diagram From Project Browser 733
  - Element To UML Model, Quick Start 53
  - Enumeration Tags To Stereotypes 1303
  - Filters To Search 627
  - Images In MDG Technology Wizard 1333
  - Instance Variable 1198
  - Key 2739
  - Learning Center In MDG Technology Wizard 1330
  - License Key 2739
  - Line Points 948
  - Linked Document Template In MDG Technology Wizard 1335
  - MDA Transforms In MDG Technology Wizard 1333
  - Model To Project 178
  - MS Word Table Of Contents 2348
  - MS Word Table Of Figures 2349
  - New Code Sections To Existing Features 1403
  - Note To Connector 946
  - Note To Link 946
  - Package In Project Browser 681
  - Package To UML Model, Quick Start 52
  - Packages To Model Document 2333
  - Pattern In MDG Technology Wizard 1328
  - Pattern To Diagram 1274, 1276
  - Port To Element 1203
  - Post To Team Review 287

- Add
  - Profile Attribute To Diagram 1280
  - Profile Connector To Diagram 1280
  - Profile Element To Diagram 1280
  - Profile In MDG Technology Wizard 1327
  - Profile Operation To Diagram 1280
  - Project Items Via Toolbar 141
  - Project Task 461
  - Property Value to Part 1201
  - RTF Report Template In MDG Technology Wizard 1334
  - Scripts In MDG Technology Wizard 1334
  - Shape Script To Stereotype In Profile 1304
  - Tagged Value Types In MDG Technology Wizard 1331
  - Tagged Values 971
  - Test Details 2214
  - Toolbox In MDG Technology Wizard 1329
  - Topic To Team Review 285
  - UML Diagram 724
  - UML Pattern To Diagram 1274, 1276
  - Views 677
- Add And Delete Attributes
  - Automation Interface Code Example 2575
- Add And Delete Methods
  - Automation Interface Code Example 2575
- Add And Manage Diagrams
  - Automation Interface Code Example 2574
- Add And Manage Elements
  - Automation Interface Code Example 2572
- Add And Manage Packages
  - Automation Interface Code Example 2571
- Add Connector
  - Automation Interface Code Example 2573
- Add Stereotypes
  - Automation Interface Code Example 2580
- Add Submenu
  - Element Context Menu, Project Browser 580
  - Package Context Menu, Project Browser 574
- Add to Project Clipboard
  - Menu Option (Edit Menu) 93
- Add-In
  - And Enterprise ArchitectDeadlocks (.NET) 2591
  - COM Interoperability 2591
  - Concurrent Method Calls 2591
  - Connect To 125
  - Create 2587
  - Create, Define Menu Items 2588
  - Deploy 2589
  - Disable 2593
  - Display Help On 125
  - Enable 2593
  - Events 2596
  - Holding State Information 2591
  - Manage 2593
  - Manager 2593
  - MDG 1289
  - Menu 125
  - Model Driven Generation 1289
  - Pre-2004 2591
  - Re-entrancy 2591
  - Register 2745
  - Run Functions From Learning Center 1353
  - Search 623, 2594
  - Search Data 2595
  - Submenu 125
  - Tasks 2586
  - Visual Basic Issues 2591
- Add-In Event
  - EA\_Connect 2597
  - EA\_Disconnect 2598
  - EA\_GetMenuItems 2598
  - EA\_GetmenuState 2599
  - EA\_MenuClick 2601
  - EA\_OnOutputItemClicked 2602
  - EA\_OnOutputItemDoubleClicked 2603
  - EA\_ShowHelp 2604
- Add-In License Management Event
  - EA\_AddinLicenseGetDescription 2607
  - EA\_AddinLicenseValidate 2606
- Add-In License Management Events
  - Add-In Model 2606
- Add-In Model
  - Add-In Event, EA\_Connect 2597
  - Add-In Event, EA\_Disconnect 2598
  - Add-In Event, EA\_GetMenuItems 2598
  - Add-In Event, EA\_GetMenuState 2599
  - Add-In Event, EA\_MenuClick 2601
  - Add-In Event, EA\_OnOutputItemClicked 2602
  - Add-In Event, EA\_OnOutputItemDoubleClicked 2603
  - Add-In Event, EA\_ShowHelp 2604
  - Add-In Event, Overview 2596
  - Add-In License Management Event, EA\_AddinLicenseGetDescription 2607
  - Add-In License Management Event, EA\_AddinLicenseValidate 2606
  - Add-In License Management Events 2606
  - Add-In Tasks 2586
  - Benefits 2585
  - Broadcast Event, EA\_FileClose 2615
  - Broadcast Event, EA\_FileNew 2615
  - Broadcast Event, EA\_FileOpen 2616

- Add-In Model
- Broadcast Event, EA\_OnPostCloseDiagram 2617
  - Broadcast Event, EA\_OnPostInitialized 2617
  - Broadcast Event, EA\_OnPostOpenDiagram 2618
  - Broadcast Event, EA\_OnPostTransform 2619
  - Broadcast Event, EA\_OnPreExitInstance 2619
  - Broadcast Event, EA\_OnRetrieveModelTemplate 2620
  - Broadcast Events 2604
  - Compartment Events 2608
  - Compartment Events, EA\_GetCompartmentData 2609
  - Compartment Events, EA\_QueryAvailableCompartments 2608
  - Context Item Events 2611
  - Context Item Events, EA\_OnContextItemChanged 2612, 2614
  - Context Item Events, EA\_OnContextItemDoubleClicked 2613
  - Create Add-In 2587
  - Create Add-In, Tricks and Traps 2591
  - Create Custom View 2670
  - Custom View 2669
  - EA\_AddinLicenseGetDescription 2607
  - EA\_AddinLicenseValidate 2606
  - EA\_Connect 2597
  - EA\_Disconnect 2598
  - EA\_FileClose 2615
  - EA\_FileNew 2615
  - EA\_FileOpen 2616
  - EA\_GetCompartmentData 2609
  - EA\_GetMenuItems 2598
  - EA\_GetMenuState 2599
  - EA\_MenuClick 2601
  - EA\_OnAttributeTagEdit 2659
  - EA\_OnConnectorTagEdit 2660
  - EA\_OnContextItemChanged 2612, 2614
  - EA\_OnContextItemDoubleClicked 2613
  - EA\_OnDeleteTechnology 2667
  - EA\_OnElementTagEdit 2661
  - EA\_OnEndValidation 2623
  - EA\_OnImportTechnology 2668
  - EA\_OnInitializeTechnologies 2663
  - EA\_OnInitializeUserRules 2622
  - EA\_OnMethodTagEdit 2662
  - EA\_OnOutputItemClicked 2602
  - EA\_OnOutputItemDoubleClicked 2603
  - EA\_OnPostActivateTechnology 2665
  - EA\_OnPostCloseDiagram 2617
  - EA\_OnPostInitialized 2617
  - EA\_OnPostNewAttribute 2639
  - EA\_OnPostNewConnector 2636
  - EA\_OnPostNewDiagram 2637
  - EA\_OnPostNewDiagramObject 2638
  - EA\_OnPostNewElement 2635
  - EA\_OnPostNewGlossaryTerm 2642
  - EA\_OnPostNewMethod 2640
  - EA\_OnPostNewPackage 2641
  - EA\_OnPostOpenDiagram 2618
  - EA\_OnPostTransform 2619
  - EA\_OnPreActivateTechnology 2664
  - EA\_OnPreDeleteAttribute 2644
  - EA\_OnPreDeleteConnector 2646
  - EA\_OnPreDeleteDiagram 2647
  - EA\_OnPreDeleteDiagramObject 2647
  - EA\_OnPreDeleteElement 2643
  - EA\_OnPreDeleteGlossaryTerm 2649
  - EA\_OnPreDeleteMethod 2645
  - EA\_OnPreDeletePackage 2648
  - EA\_OnPreDeleteTechnology 2666
  - EA\_OnPreExitInstance 2619
  - EA\_OnPreNewAttribute 2655
  - EA\_OnPreNewConnector 2652
  - EA\_OnPreNewDiagram 2653
  - EA\_OnPreNewDiagramObject 2654
  - EA\_OnPreNewElement 2651
  - EA\_OnPreNewGlossaryTerm 2658
  - EA\_OnPreNewMethod 2656
  - EA\_OnPreNewPackage 2657
  - EA\_OnRetrieveModelTemplate 2620
  - EA\_OnRunAttributeRule 2627
  - EA\_OnRunConnectorRule 2626
  - EA\_OnRunDiagramRule 2625
  - EA\_OnRunElementRule 2624
  - EA\_OnRunMethodRule 2628
  - EA\_OnRunPackageRule 2625
  - EA\_OnRunParameterRule 2629
  - EA\_OnStartValidation 2623
  - EA\_QueryAvailableCompartments 2608
  - EA\_ShowHelp 2604
  - Interface 2585
  - Introduction 2585
  - MDG Add-Ins 2671
  - MDG Add-Ins, MDG Events 2672
  - MDG Events, MDG\_BuildProject 2673
  - MDG Events, MDG\_Connect 2674
  - MDG Events, MDG\_Disconnect 2675
  - MDG Events, MDG\_GetConnectedPackages 2676
  - MDG Events, MDG\_GetProperty 2676
  - MDG Events, MDG\_Merge 2678
  - MDG Events, MDG\_NewClass 2680
  - MDG Events, MDG\_PostGenerate 2681

## Add-In Model

MDG Events, MDG\_PostMerge 2682  
 MDG Events, MDG\_PreGenerate 2683  
 MDG Events, MDG\_PreMerge 2683  
 MDG Events, MDG\_PreReverse 2684  
 MDG Events, MDG\_RunExe 2685  
 MDG Events, MDG\_View 2686  
 Model Validation Broadcasts 2621  
 Model Validation Broadcasts,  
 EA\_OnEndValidation 2623  
 Model Validation Broadcasts,  
 EA\_OnInitializeUserRules 2622  
 Model Validation Broadcasts,  
 EA\_OnRunAttributeRule 2627  
 Model Validation Broadcasts,  
 EA\_OnRunConnectorRule 2626  
 Model Validation Broadcasts,  
 EA\_OnRunDiagramRule 2625  
 Model Validation Broadcasts,  
 EA\_OnRunElementRule 2624  
 Model Validation Broadcasts,  
 EA\_OnRunMethodRule 2628  
 Model Validation Broadcasts,  
 EA\_OnRunPackageRule 2625  
 Model Validation Broadcasts,  
 EA\_OnRunParameterRule 2629  
 Model Validation Broadcasts,  
 EA\_OnStartValidation 2623  
 Model Validation Example 2630  
 Post-New Events 2634  
 Post-New Events, EA\_OnPostNewAttribute  
 2639  
 Post-New Events, EA\_OnPostNewConnector  
 2636  
 Post-New Events, EA\_OnPostNewDiagram  
 2637  
 Post-New Events, EA\_OnPostNewDiagramObject  
 2638  
 Post-New Events, EA\_OnPostNewElement  
 2635  
 Post-New Events, EA\_OnPostNewGlossaryTerm  
 2642  
 Post-New Events, EA\_OnPostNewMethod  
 2640  
 Post-New Events, EA\_OnPostNewPackage  
 2641  
 Pre-Deletion Events 2642  
 Pre-Deletion Events, EA\_OnPreDeleteAttribute  
 2644  
 Pre-Deletion Events, EA\_OnPreDeleteConnector  
 2646  
 Pre-Deletion Events, EA\_OnPreDeleteDiagram  
 2647

Pre-Deletion Events,  
 EA\_OnPreDeleteDiagramObject 2647  
 Pre-Deletion Events, EA\_OnPreDeleteElement  
 2643  
 Pre-Deletion Events,  
 EA\_OnPreDeleteGlossaryTerm 2649  
 Pre-Deletion Events, EA\_OnPreDeleteMethod  
 2645  
 Pre-Deletion Events, EA\_OnPreDeletePackage  
 2648  
 Pre-New Events 2650  
 Pre-New Events, EA\_OnPreNewAttribute 2655  
 Pre-New Events, EA\_OnPreNewConnector  
 2652  
 Pre-New Events, EA\_OnPreNewDiagram 2653  
 Pre-New Events, EA\_OnPreNewDiagramObject  
 2654  
 Pre-New Events, EA\_OnPreNewElement 2651  
 Pre-New Events, EA\_OnPreNewGlossaryTerm  
 2658  
 Pre-New Events, EA\_OnPreNewMethod 2656  
 Pre-New Events, EA\_OnPreNewPackage 2657  
 Search Data, XML Format 2595  
 Tagged Value Broadcast Events 2658  
 Tagged Value Broadcast Events,  
 EA\_OnAttributeTagEdit 2659  
 Tagged Value Broadcast Events,  
 EA\_OnConnectorTagEdit 2660  
 Tagged Value Broadcast Events,  
 EA\_OnElementTagEdit 2661  
 Tagged Value Broadcast Events,  
 EA\_OnOperationTagEdit 2662  
 Technology Event, EA\_OnInitializeTechnologies  
 2663  
 Technology Events 2663  
 Technology Events, EA\_OnDeleteTechnology  
 2667  
 Technology Events, EA\_OnImportTechnology  
 2668  
 Technology Events,  
 EA\_OnPostActivateTechnology 2665  
 Technology Events, EA\_OnPreActivateTechnology  
 2664  
 Technology Events, EA\_OnPreDeleteTechnology  
 2666  
 Additional Collection Class 1906  
 Administrator  
 Security Permissions 253  
 Advanced  
 Settings, Connector 968  
 Settings, Elements 847  
 Tag Management 975  
 Advanced (Element)  
 Submenu 109

- Advanced Options
  - RTF Report 2261
- Advanced Search Options 626
- Aggregate
  - Connector 1208
  - Relationship 1208
- Aggregation Connector
  - Change Form 1209
- Align
  - Elements From Toolbar 145
  - Multiple Elements 809
- Alignment
  - Submenu 112
- All User Permissions
  - Dialog, User Security 261
  - View, User Security 261
- Allocated Resources
  - View, Project Calendar 510
- Allocated Work
  - Tab, Personal Information Window 493
- Alternate Path
  - Add To Scenario (Toolbar) 857
- Alternate Path, Scenario 852
- Alternative Image
  - For Element 755
  - In Diagram 755
  - Select 755
  - Stereotype 1269
- Analysis
  - Elements and Connectors 715
  - Group, Toolbox 715
  - Models 1503
  - Models, Business Processes 1517
  - Stereotypes 1508
- Analysis Diagram
  - Description 1509
  - Diagram 1509
  - Elements And Connectors 1509
  - Example 1511
- Analyzer
  - Menu 120
- Analyzer Script 1861
  - Options, Filters 1863
- Analyzer Scripts
  - Add New 1828
  - Build 1828
  - Clean 1828
  - Copy 1828
  - Create 1828
  - Debug 1828
  - Delete 1828
  - Deploy 1828
  - Edit 1828
  - Export 1828
  - Import 1828
  - Manage, Visual Execution Analyzer 1828
  - Paste 1828
  - Rebuild 1828
  - Rename 1828
  - Run 1828
  - Set As Default 1828
- and any breakpoints should 1853
- Android Mobile
  - Application Pattern 1738
- ANSI C 1911
  - Modeling Conventions 1745
- Anti-Aliased Rendering
  - Of Diagrams 550
- Anti-Aliased Text
  - In Diagrams 550
- Apache Tomcat
  - Server Configuration 1849
  - Server, Debugging 1846
  - Service Configuration 1850
- App Object
  - Automation Interface 2412
- APPDATA
  - Import MDG Technologies 1288
- Appearance
  - Apply From Clipboard 837
  - Connectors Context Menu Section 941
  - Copy For Element 837
  - Default, Of Element 817
  - Element Context Menu 837
- Appearance (Element)
  - Autosize 112
  - Submenu 112
- Applets
  - Java, In Internet Browsers, Debug 1845
- Application
  - Connector 714
- Application Look Dialog 162
- Application Pattern
  - Android Mobile 1738
  - ASP.NET Web Service 1738
  - Generate 1738
  - Java Program 1738
  - MFC Windows 1738
- Application Workspace 86
- Apply
  - Rigorous Security Mode Lock 274
  - RTF Report Filter (Legacy) 2320
  - Stereotype To Dependency Relationship 1221
  - Stereotype To Element 1265

- Apply
  - Stereotype To UML Construct 1265
  - User Lock 274
- ArcGIS
  - Code Generation Language Options 1910
  - Connectors 1634
  - Disable 1633
  - Elements 1634
  - Export XML Workspace 1640
  - Import from XMI 417
  - Import XML Workspace 1641
  - MDG Technology For 1633
  - Options 1910
  - Profile 1633
  - Relationships 1634
  - Schema, Export 1640
  - Schema, Import 1641
  - Set Coordinate Systems 1637
  - Support For In Enterprise Architect 1633
  - Supported Types 1634
  - Toolbox Pages 1634
- ArchiMate
  - Concept 1621
  - Diagram 1621
  - Disable 1621
  - Elements 1621
  - Enable 1621
  - MDG Technology 1621
  - Relationships 1621
  - Toolbox Pages 1621
- Argument
  - Call, Synchronize With Behavior Parameter 911
  - For Behavioral Parameter 912
  - Invocation, Synchronize With Behavior Parameter 911
- Arrange
  - Connectors 946
- Array
  - View Elements Of, Debugger 1881
- Arrow
  - Quick Linker 791
- Artifact
  - Element 1179, 1180
- ASA
  - Data Repository, Connect To 219
  - ODBC Driver, Set Up 205
  - Repository, Create 198
  - Upsize To 234
- ASP .NET
  - Debug 1853
- ASP.NET
  - Application Pattern 1738
- Assembly
  - Connector 1209
  - Relationship 1209
- Assign
  - Information to Tagged Values 973
  - People To Changes 2247
  - People To Defects 2247
  - Tagged Values To Item 972
- Assign Activity
  - BPEL 2.0, Model 1613
- Assignment
  - BPEL 1.1, Create 1589
- Associate
  - Calls With Behaviors 911
  - Connector 1210
  - Invocations With Behaviors 911
  - Relationship 1210
- Associated Files
  - Elements 875
- Association 961
  - Class 1673
  - Connector 1210
  - Connector, Set Collection Class 1904
  - Details 961
  - Dot On Connector 964
  - N-Ary 1673
  - Properties 961
  - Relationship 1210
  - Set Derived Property 939
  - Specialisation, Set Up 958
- Association Class
  - Connector 1215
  - Find In Project Browser 938
  - Link New Class To Association 1216
  - Locate In Project Browser 1215
  - Relationship 1215
  - Shape Scripts 1304
- Association End
  - Qualifiers 1211, 1212
- Asynchronous Signal Message
  - Associate With Signal 1241
  - Connector 1241
  - Relationship 1241
- Attach
  - Note To Link 946
- Attach To Process Dialog 1888
- Attribute
  - Add And Delete, Automation Interface Code Example 2575
  - Add To Element, In-place Editor 924
  - Automatic Properties, C# 893
  - Collections 889

- Attribute
  - Connect To 944
  - Constraints 890
  - Context Menu, Project Browser 582
  - Copy Between Elements 823
  - Create 885
  - Create Fast, Option 1899
  - Create Object From 896
  - Definition 885
  - Delete 887
  - Delete If Not In Code In Reverse Synchronization 1899
  - Derived 887
  - Dialog 885
  - Dialog, Constraints Tab 890
  - Dialog, Detail Tab 889
  - Dialog, General Tab 887
  - Disconnect From 944
  - Edit Keyword 921
  - Edit Name, In-Place Editor 918
  - Edit Scope 920
  - Edit Stereotype, In-place Editor 919
  - Fast Create 885
  - Field Substitution Macros 1414
  - Imported, Default Name Generated From 1899
  - Inherited, Display 895
  - Inherited, Show 745
  - Introduction 885
  - Link To Element Via Object 896
  - Message Part, WSDL 2074
  - Modify 885
  - Move Between Elements 824
  - Multiplicity 889
  - Of Toolbox Page 1339
  - PData & StyleEx, Diagram Profiles 1348
  - Private, Icon 585
  - Properties, Create 892
  - Protected, Icon 585
  - Qualifiers 889, 1212
  - Redefine Property 889
  - Show On Diagram 745
  - Stereotyped, For Columns 1942
  - Subsetted Property 889
  - Supported, Create Composite Elements 1314
  - Supported, Define Child Diagram Types 1315
  - Supported, Metatype In Profiles 1308
  - Supported, Stereotype In Profiles 1308
  - Tagged Values 891
  - UML Property, isUnique 889
  - Work With, Automation Interface Code Example 2581
- Attribute Class
  - Alias, Automation Interface 2392
  - Automation Interface, ElementFeatures Package 2392
  - AttributeConstraint
    - Automation Interface, ElementFeatures Package 2396
  - AttributeTag
    - Automation Interface, ElementFeatures Package 2397
  - Audit
    - Requirements 1491
    - Scope 384
  - Audit History Tab
    - Description 392
    - How To Display 392
    - On Output Window 164, 592
  - Audit Options
    - All 384
    - Connectors Audited 384
    - Core Structural 384
    - Custom 384
    - Elements Audited 384
    - Maintenance 384
  - Audit View
    - Advanced Mode 389
    - Audit Changes 389
    - Controls 389
    - Custom Time Periods 389
    - Deleted Mode 389
    - Display Database Changes 389
    - Filter By Time 389
    - Performance Problems 394
    - Raw Mode 389
    - Refresh 389
    - Search 389
    - Slow Loading 394
    - Slow Navigation 394
    - Sort 389
    - Standard Mode 389
  - Auditing
    - Alternative To Differencing 394
    - And Performance Of Enterprise Architect 382
    - And RTF Reporting 382
    - Audit Tree 387
    - Clear Logs 384
    - Display Audit Results 387
    - Enable 384
    - How To Invoke 383
    - Include Reverse Engineering 384
    - Include XMI Export 384
    - Include XMI Import 384
    - Introduction 382

- Auditing
  - Large Deletion Issue 393
  - Level, Core 384
  - Level, Extended 384
  - Level, Standard 384
  - Load Logs 384
  - Performance Issues 393
  - Quick Start 383
  - Record Display 387
  - Reverse Engineering Issue 393
  - Save Logs 384
  - Settings 384
  - Use Database Timestamp 384
  - View 387
  - XMI Import Issue 393
- Authentication
  - Accept Windows Authentication 256
  - Automatic Delete On Relocation Of Project 256
- Author
  - Attributes 2425
  - Define 989
  - From Windows Active Directory 989
  - Methods 2425
- Author Collection
  - Automation Interface Repository 2425
- Auto Counter
  - Of Elements, Set Up 799
- Auto Increment Column Values
  - Impact Of Replication 536
  - Impact Of XMI Import 536
  - Reset 536
- Auto Naming
  - And Requirements 1473
  - Maintenance Items 2237
  - Of Element, Set Up 799
  - Task Items 461
  - Test Items 2213
- Auto Numbering
  - And Requirements 1473
  - Apply 463
  - Maintenance Items 2237
  - Of Elements, Set Up 799
  - Task Items 461
  - Test Items 2213
- Auto Route Diagram Layout 786
- Auto Save
  - Diagram Changes 553
- Autocompletion List
  - Code Editor, Common 1812
- Autohide
  - Reveal Autohidden Window 137
  - Turn Off 137
  - Turn On 137
  - Windows 137
- Autolayout
  - Diagram 787
  - Options 787
- Automatic
  - Save Of Diagram Changes 553
- Automatic Exclusive Edit Lock 253
- Automatically Hidden Windows
  - Animate 98
- Automation
  - Of Enterprise Architect 2371
- Automation Interface 2410
  - App Object 2412
  - Attribute Class, ElementFeatures Package 2392
  - AttributeConstraint, ElementFeatures Package 2396
  - AttributeTag, ElementFeatures Package 2397
  - Available Resources 2389
  - Call Executables From Enterprise Architect 2381
  - Call From Enterprise Architect 2387
  - Code Example, Add And Delete Attributes 2575
  - Code Example, Add And Delete Methods 2575
  - Code Example, Add And Manage Diagrams 2574
  - Code Example, Add And Manage Elements 2572
  - Code Example, Add And Manage Packages 2571
  - Code Example, Add Connector 2573
  - Code Example, Add Stereotypes 2580
  - Code Example, Iterate Through EAP File 2570
  - Code Example, Open The Repository 2570
  - Code Example, Use Element Extras 2575
  - Code Example, Use Repository Extras 2578
  - Code Example, Work With Attributes 2581
  - Code Example, Work With Methods 2582
  - Code Examples, Introduction 2569
  - Connect From Borland Delphi 7.0 2382
  - Connect From Java 2382
  - Connect From MS C# 2382
  - Connect From MS Visual Basic 6.0 2382
  - Connect To 2381
  - Connector Package Diagram 2507
  - Connector, Connector Package 2508
  - ConnectorConstraint, Connector Package 2514
  - ConnectorEnd, Connector Package 2515
  - ConnectorTag, Connector Package 2518
  - ConstLayoutStyles Enum 2414
  - Constraint, Element Package 2476



- Automation Interface 2410
  - CreateBaselineFlag Enum 2416
  - CreateModelType Enum 2416
  - CustomProperties Collection, ElementFeatures Package 2399
  - Diagram Package 2522
  - Diagram, Diagram Package 2523
  - DiagramLinks, Diagram Package 2528
  - DiagramObject, Diagram Package 2529
  - Document Generator Interface 2561
  - DocumentBreak Enum 2416
  - DocumentGeneratorClass 2562
  - DocumentType Enum 2416
  - EAEditionTypes Enum 2417
  - Effort, Element Package 2477
  - Element Package Diagram 2474
  - Element Package, File 2491
  - Element, Element Package 2478
  - ElementFeatures Package Diagram 2391
  - EmbeddedElements Collection, ElementFeatures Package 2399
  - Enumerations 2413
  - EnumRelationSetType Enum 2417
  - Examples 2381
  - Examples and Tips 2386
  - ExportPackageXMIFlag Enum 2418
  - Introduction 2381
  - Issue, Element Package 2492
  - Mail Interface Package 2566
  - MailInterface Class, Mail Interface Package 2566
  - MDGMenus Enum 2418
  - MessageFlag 2419
  - Method, ElementFeatures Package 2400
  - MethodConstraint, ElementFeatures Package 2404
  - MethodTag, ElementFeatures Package 2405
  - Metric, Element Package 2494
  - Model 2390
  - ObjectType Enum 2419
  - Package 2390
  - Parameter, ElementFeatures Package 2406
  - ParamTag, ElementFeatures Package 2408
  - Partitions Collection, ElementFeatures Package 2409
  - Project Interface 2534
  - Project, Project Interface 2535
  - Properties, ElementFeatures Package 2410
  - Property ElementFeatures Package 2410
  - PropType Enum 2421
  - Reference 2389
  - ReloadType Enum 2421
  - Repository 2448
  - Repository Package 2423, 2433
  - Repository, Author Collection 2425
  - Repository, Client Collection 2425
  - Repository, Collection Class 2427
  - Repository, Datatype 2429
  - Repository, EventProperties 2431
  - Repository, EventProperty 2432
  - Repository, ModelWatcher 2432
  - Repository, ProjectIssues 2442
  - Repository, ProjectResource 2444
  - Repository, ProjectRole 2445
  - Repository, PropertyType 2446
  - Repository, Reference 2447
  - Repository, Stereotype 2470
  - Repository, Task 2471
  - Repository, Term 2473
  - Requirement, Element Package 2495
  - Resource, Element Package 2497
  - Risk, Element Package 2498
  - RoleTag, Connector Package 2520
  - Scenario, Element Package 2499
  - ScenarioDiagramType Enum 2422
  - ScenarioExtension, Element Package 2501
  - ScenarioStep, Element Package 2502
  - ScenarioStepType Enum 2422
  - ScenarioTestType Enum 2423
  - Set Up Visual Basic 2385
  - Swimlane, Diagram Package 2534
  - SwimlaneDef, Diagram Package 2531
  - Swimlanes, Diagram Package 2532
  - TaggedValue, Element Package 2504
  - Test, Element Package 2506
  - Transitions Collection, ElementFeatures Package 2412
  - Using 2382
  - VB GetObject Support 2412
  - XMIFlag Enum 2423
- Autosize
  - Element, Single 750
  - Elements, As Group 750
- Available Resources
  - Automation Interface 2389
- B -**
- Base Project
  - Copy 190
  - New 190
- Baseline
  - And Differences, Overview 394
  - Considerations 396

- Baseline
  - Create 400
  - Delete 398
  - Diagram Comparison 403
  - Export 398
  - Import 398
  - Include XMI Stubs 400
  - Load From Alternative Model 398
  - Manage 398
  - Merge With Current Model, Overview 394
  - Model 396
  - Overview 396
  - Requirements 1491
  - Scenarios 396
  - Versions 400
- Baseline Diagram Compare
  - Dialog 403
- BaseModel Script
  - InnoDB 224
  - MyISAM 224
- Basic Path, Scenario 852
- Batch Generate
  - Elements With Code 111
  - RTF Resource Documents, Automatically 2259
  - RTF Resource Documents, Manually 2259
- Batch XMI
  - Export 430
  - Import 431
- Behavior
  - Associate With Calls 911
  - Associate With Invocations 911
  - Instance 1260
  - Of Operation, Display In Diagram 904
  - Operation, Associate With 904
  - Parameter, Tagged Values 914
  - Parameters, Define 913
  - Synchronize Call Arguments With Parameters 911
  - Synchronize Invocation Arguments With Parameters 911
- Behavior Call
  - Actions For RuleTask Operations 1534
  - Interaction Occurrence 910
  - Modeling 910
  - Parameter Arguments 912
- Behavioral Diagram
  - Elements 1090
  - Overview 1027
- Behavioral Model Templates 1449
- Behavioral Modeling 899
- Behavioral Models
  - Generate Code From 1778
- Behavioral Parameter
  - Edit 912
  - Extend 912
  - Reassign 912
  - Set 912
- Behaviors
  - Of Activities 910
  - Of Interactions 910
- Bend
  - Connector At Cursor 948
- Bend Connector 55
- Bezier Lines 948
- Binary Module
  - Import, Reverse Engineering 1797
  - Languages Supported 1797
- Bind
  - Breakpoints 1873
  - Fail To, Breakpoints 1873
- Binding
  - In Connector 964
  - WSDL Diagram 2078
  - WSDL Element 2078
- Binding Class
  - And Binding Connector 1257
- Binding Expression
  - Binding Connector 1257
- Bitmap Image
  - In Diagrams 755
- Blue Exclamation Mark 276
- Bookmark
  - Clear 93
  - Clear All 93
  - Code Editor, Common 1810
  - For RTF Report 2346
  - Insert In RTF Template 2291
  - Multiple Elements 530
  - Package As 2346
  - RTF, In Master Document Element 2327
  - Selected Element 93
  - Triangle 530
- Bookmark Selected
  - Menu Option (Edit Menu) 93
- Border
  - Red 276
- Boundary
  - Element 1663
  - Element Settings 1171
  - Element, Create 1664
  - Object Settings 1171
  - Properties 1171
- Box Diagram Layout 775
- BPEL

## BPEL

1.1 Assignments, Create 1589  
 1.1 Create Model 1573  
 1.1 End Event, Model 1586  
 1.1 Example 1571  
 1.1 Generate 1594  
 1.1 Intermediate Event, Model 1578  
 1.1 Model 1571  
 1.1 Model, Create 1573  
 1.1 Modeling Restrictions 1571  
 1.1 Package Structure 1573  
 1.1 Package Structure, Create 1571  
 1.1 Process, Capture Via BPMN 1.1 1571  
 1.1 Process, Properties 1576  
 1.1 Start Event, Model 1577  
 1.1, BPMN Elements Not Mapable 1576  
 1.1, Create Assignments 1589  
 1.1, Looping Constructs 1581  
 1.1, Model A Process 1576  
 1.1, Model Activity 1581  
 1.1, Model Gateway 1584  
 1.1, Model Pool 1588  
 1.1, Model Sequence Flow Connector 1587  
 1.1, Sub Process Types 1581  
 1.1, Task Types 1581  
 1.1, Web Service, Create 1591  
 2.0 Assign Activity, Model 1613  
 2.0 End Event, Model 1608  
 2.0 Example 1595  
 2.0 Intermediate Event, Model 1602  
 2.0 Looping Constructs 1605  
 2.0 Model 1595  
 2.0 Model Activity 1605  
 2.0 Model Gateway 1607  
 2.0 Model, Create 1597  
 2.0 Modeling Restrictions 1595  
 2.0 Package Structure 1597  
 2.0 Package Structure, Create 1595  
 2.0 Process, Capture Via BPMN 2.0 1595  
 2.0 Process, Model 1599  
 2.0 Sub Process Types 1605  
 2.0 Task Types 1605  
 2.0 WSO, Create 1615  
 2.0, Generate Code 1618  
 2.0, Model Data Object 1610  
 2.0, Model Property 1611  
 2.0, Model Sequence Flow Connector 1612  
 2.0, Start Event, Model 1601  
 2.0, Use Of BPMN Pool 1612  
 Concepts 1570  
 Create 2.0 Model 1597  
 Diagram 1570

Disable 1570  
 In Enterprise Architect 1570  
 MDG Technology 1570  
 Model Validation 1619  
 Modeling Restrictions 1570  
 Process Element, BPMN 1.1 1547  
 Toolbox, Via BPMN 1.1 1571  
 Toolbox, Via BPMN 2.0 1595

## BPMN

1.0 1545  
 1.0 Elements 1547  
 1.0 Relationships 1547  
 1.0 Toolbox Pages 1547  
 1.1 1545  
 1.1 Elements 1547  
 1.1 Relationships 1547  
 1.1 Toolbox Pages 1547  
 2.0 1545  
 2.0 Business Process Elements 1550  
 2.0 Business Process Relationships 1550  
 2.0 Business Process Toolbox Pages 1550  
 2.0 Choreography Elements 1553  
 2.0 Choreography Relationships 1553  
 2.0 Choreography Toolbox Pages 1553  
 2.0 Collaboration Elements 1555  
 2.0 Collaboration Relationships 1555  
 2.0 Collaboration Toolbox Pages 1555  
 2.0 Conversation Elements 1557  
 2.0 Conversation Relationships 1557  
 2.0 Conversation Toolbox Pages 1557  
 2.0 Elements 1549  
 2.0 Model 1569  
 2.0 Relationships 1549  
 2.0 Toolbox Pages 1549  
 2.0 Type Elements 1560  
 2.0 Type Relationships 1560  
 2.0 Type Toolbox Pages 1560  
 2.0 XML 1569  
 Change Element Appearance 1561  
 Concepts 1545  
 Core Page 1547  
 Diagram 1545  
 Disable 1545  
 Element Appearance, Change 1561  
 MDG Technology 1545  
 Migrate 1.0 Model To 1.1 1565  
 Migrate 1.1 Model To 2.0 1567  
 Serializing 2.0 In XML 1569  
 Toolbox Page 1545  
 Types Page 1547  
 Update Version Via Toolbox Element 1561  
 Version Differences In Appearance 1561

- BPMN 2.0
  - Activity Types 1605
  - End Event, Types 1608
  - Gateway Types 1607
  - Intermediate Event, Types 1602
  - Sub-Process Types 1605
  - Task Types 1605
- BPMN 1.1
  - Activity Types 1581
  - BPEL 1.1 Sequence Flow 1587
  - Elements Not Mappable to BPEL 1.1 1576
  - End Event, Types 1586
  - Gateway Types 1584
  - Intermediate Event, Types 1578
  - Pool 1588
  - Start Event, Types 1577
  - Sub-Process Types 1581
  - Task Types 1581
- BPMN 2.0
  - BPEL Sequence Flow 1612
  - Data Object 1610
  - Property 1611
  - Start Event, Types 1601
- Brace Matching
  - Code Editor, Common 1810
- Branching Macros
  - Code Template Syntax 1446
- Breakpoint
  - Delete 1869
  - Difference From Recording Marker 2155
  - Disable 1869
  - Enable 1869
  - Fail To Bind 1873
  - Failure Message 1869
  - Management 1869
  - Marker On Element During Simulation 2114
  - Model Simulation, Using 2114
  - Properties 1871
  - Set For Modifiable Data 1882
  - Set In Code 1871
  - Sets, Model Simulation 2114
  - States 1869
- Breakpoints & Events Window
  - Simulation Events Tab 2125
- Breakpoints And Markers Window 1869, 2157
- Broadcast Event
  - Add-In Model 2604
  - EA\_FileClose 2615
  - EA\_FileNew 2615
  - EA\_FileOpen 2616
  - EA\_OnPostCloseDiagram 2617
  - EA\_OnPostInitialized 2617
  - EA\_OnPostOpenDiagram 2618
  - EA\_OnPostTransform 2619
  - EA\_OnPreExitInstance 2619
  - EA\_OnRetrieveModelTemplate 2620
- BroadcastSignal Action
  - Signal Tab 1095
- Browser
  - Element 876
- Build
  - Application 1866
  - Application, Analyzer menu Option 120
  - Clean Commands, Add 1833
  - Code 1824
  - Execute Command 1866
  - Locate Compiler Error In Code 1866
  - Output 1866
- Build And Run Submenu
  - Package Context Menu, Project Browser 576
- Build Script
  - Create 1832
  - Deploy Script, Create 1862
  - Introduction 1832
  - Recursive 1832
  - Run Script, Create 1862
  - Test Script, Create 1834
- Build Systems Using UML
  - Enterprise Architect 9
- Built-In
  - Diagram Types 1348
  - Transformations 1686
- Business
  - Models 1512
  - Scenarios & Requirements 879
- Business Analyst
  - And Enterprise Architect 74
  - Project Role 74
- Business and Software Engineering Edition
  - Of Enterprise Architect 22
- Business Domain Model
  - Create 1529
- Business Interaction Diagram
  - Description 1514
  - Elements And Connectors 1514
  - Example 1514
- Business Model
  - Analysis 1517
- Business Modeling
  - Business Process Outline 1522
  - Events 1520
  - Example 1515
  - Goals 1521
  - Information 1519

- Business Modeling
    - Inputs 1519
    - Outputs 1521
    - Process Element 1518
    - Process Modeling Notation 1518
    - Processes 1515
    - Resources 1519
    - Traceability 635
  - Business Modeling Diagram
    - Description 1514
    - Elements And Connectors 1514
    - Example 1514
  - Business Process
    - Analysis 662, 665
    - Model Template 662
    - Model, Template 665
    - Outline 1522
  - Business Process Execution Language (BPEL) 1570
  - Business Process Modeling 1509
  - Business Process Modeling Notation (BPMN) 1545
  - Business Rule
    - Add To Rule Task 1536
    - Create 1536
    - Element 1527
    - Export To CSV File 1536
    - Generate Code 1543
    - Modeling 1527
    - Parameters As Variables 1533
    - Remove From Rule Task 1536
    - Validate 1542
  - Business Rule Modeling
    - Business Domain Model, Create 1529
    - Business Rules 1527
    - Compose Rules 1536
    - Computation Rule Table 1536
    - Decision Table 1536
    - Export Rules To CSV File 1536
    - Generate Code 1543
    - In Class Operations 1534
    - In RuleTask Actions 1527
    - Overview 1522
    - Rule Composer 1536
    - Rule Flow Diagram 1530
    - Rule Model 1527
    - Vocabulary 1529
  - Import, Reverse Engineering 1793
  - Modeling Conventions 1745, 1747
  - Object Oriented Programming 1747
  - Options 1911
  - C#
    - Code Generation Language Options 1912
    - Import, Reverse Engineering 1793
    - Modeling Conventions 1748
    - Options 1912
    - Transformation 1687
  - C++
    - Code Generation 1912
    - Debug Symbols 1842
    - Debugging, General Setup 1841
    - Implementation Files 1912
    - Import, Reverse Engineering 1793
    - Language Options 1912
    - Modeling Conventions 1750
    - Modeling Conventions, CLI Extensions 1753
    - Modeling Conventions, Managed 1752
    - Sample Model, Visual Execution Analysis 2145
    - Set Up Debug Session 1841
  - Calendar
    - Allocated Resources View 510
    - Calendar Panel 502
    - Calendar Panel, Configure Event Subtypes 509
    - Diary Panel 502
    - Display Events 505
    - Options 502
    - Overview 502
    - Project 502
    - Project CalendarView 505
    - Project Tasks View 513
    - Recurring Events 505
    - Toolbar 502
  - Calibration
    - Of Project Factors 516
  - Call
    - Associate With Behaviors 911
    - Automation Interface From Enterprise Architect 2387
    - Re-associate With Behavior 911
    - Self Message 1236
    - Synchronize Arguments With Behavior Parameters 911
  - Call Action
    - Pin As Argument 1097
  - Call Actions 1787
  - Call Stack
    - Create Sequence Diagram 1886
    - Options 1884
    - View 1884
- C -**
- C
- Code Generation Language Options 1911
  - Debugging, General Setup 1841

- Call Stack
  - Window 1884
- CallOperationAction 1787
- Calltips
  - Code Editor, Common 1812
- Camel Case
  - Naming Format 1728
- Cancel
  - Default Diagram, Model 744
  - Validation 102
- Capture State Changes
  - Setup To, Visual Execution Analyzer 2166
- Cardinality (Multiplicity)
  - Define 986
- CASE Tool
  - Enterprise Architect 3
- Category
  - Add To Team Review 284
  - Create 284
  - Delete 280
- Central Buffer Node
  - Element 1107
- Chaining Transformations 1684
- Change
  - BPMN Element Appearance 1561
  - Connector Source Or Target 947
  - Connector Type 947
  - Diagram Type 740
  - Element 2244
  - Element Type 109, 809
  - Elements And Requirements 1491
  - Form Of Aggregation Connector 1209
  - Tracking 381, 382
- Change Conflicts
  - Resolve 250
- Change Element
  - Hide Stereotype Letter 2246
  - Show Stereotype Letter 2246
- Change Management
  - And Requirements 1491
  - Auditing 311
  - Baselines And Differences 311
  - In Enterprise Architect 311
  - Introduction 311
  - Model Transfer 411
  - Project Data Transfer 311
  - Tracking Changes 381
  - Version Control 311, 312
- Character Set
  - Set Up For RTF Report 2269
  - Set Up For RTF Report (Legacy) 2326
- Check
  - Data Integrity 535
  - Model Integrity 535
  - Project Integrity 535
- Check Constraint
  - Create 1966
  - What Is A? 1966
- Check In
  - Explanation 357
  - Offline Packages 322
- Check Out
  - Explanation 357
  - Packages Offline 322
  - Revision Of Package 376
- Checked In Package
  - Icon 585
- Checked Out Package
  - Icon 585
- Child
  - Confirm Element As Parent 805
- Child Diagram
  - Of Composite Element 825
- Child Element
  - Paste Object As 733
- Choice
  - Element 1108
- Circle Diagram Layout 770
- Circle Notation
  - For Interface Element 1194
  - For Shaped Elements 1177
- Class
  - Active Classes 1182
  - Collection, Set 1904
  - Created In Transformation, Connect To 1724
  - Element 1181
  - Elements And Connectors 708
  - Elements, Imported 1976
  - Field Substitution Macros 1416
  - Group, Toolbox 708
  - Make Into Association Class 1216
  - Members, Show/Hide On Diagram 732
  - Model Template 662, 668
  - Parameterized Classes (Templates) 1183
  - Partial 1748
  - Partial, Generate 1772
  - Reset Options 1922
  - Show Realised Interfaces On Diagram 760
  - Source Code Generation 1768
  - View 676
- Class Diagram
  - Description 1014
  - Elements And Connectors 1014
  - Example 1015

- Class Invariant Dialog 2193
- Class Markup Selection Dialog 2160
- CLASSGUID
  - Add-In Hidden Field 2594
- Classifier
  - Behavior 1137
  - Drop As Link 733
  - Drop As New Instance 733
  - Item Conveyed 1228
  - Of Objects 898
  - Properties 1019
  - Set 899
  - Use 899
- CLASSTYPE
  - Add-In Hidden Field 2594
- Clean
  - Data 535
  - Project 535
- Clean Commands
  - Add To Build Script 1833
- Clean Project
  - Prior To Build 1833
- Clear All Bookmarks
  - Menu Option (Edit Menu) 93
- Clear Project Clipboard
  - Menu Option (Edit Menu) 93
- Clear Selection
  - Menu Option (Edit Menu) 93
- CLI Extensions
  - C++ Modeling Conventions 1753
- Client
  - Define 993
- Client Collection
  - Automation Interface Repository 2425
- Clipboard
  - Copy Team Review Path To 280
- Clipboard File Format
  - Define 544
- Close
  - Full Screen 129
- Close Project
  - Menu Option (File Menu) 92
- CLR Versions
  - Debug .NET 1851
- Code
  - Breakpoint, Set 1871
  - Build 1824
  - Debug 1824
  - Debug, Step Into Function Calls 1875
  - Debug, Step Out Of Functions 1875
  - Debug, Step Over Lines Of Code 1875
  - Debug, Step Through Function Calls 2162
  - Delete From Features In Model In Fwd Synchronization 1899
  - Generate For Business Rule 1543
  - Generated From State Machine 1781
  - Generation, Toolbar 142
  - Import, Select Language 142
  - Language, Set Default 142
  - Synchronize 1401
- Code Breakpoint
  - Set 1871
- Code Editor
  - Compare 1801
  - External 1801
  - Functions, Editing 1809
  - Functions, Overview 1809
  - Internal (External Code) 1801
  - Languages Supported 1800
  - Supported Languages 1800
  - Variants 1801
  - View 1801
  - Window 1801
- Code Editor, Common
  - Autocompletion List 1812
  - Automatic Indentation 1810
  - Bookmarks 1810
  - Brace Matching 1810
  - Calltips 1812
  - Commenting Selections 1810
  - Configure Search Options 1805
  - Context Menu 1805
  - Create Use Case For Method Option 1807
  - Cursor History 1810
  - Customize 134
  - Debug Tooltips 1879
  - Find And Replace 1814
  - Functions 134
  - IME 1805
  - Input Method Editor 1805
  - Intellisense 1812
  - Key Bindings 1819
  - Line Selection 1810
  - Manual Indentation 1810
  - Mouseovers 1812
  - Overview 134
  - Scope Guides 1810
  - Search Facility 1805
  - Syntax Highlighting 1810
  - Tooltips, Debug 1879
  - Zooming 1810
- Code Engineering
  - And MDG Integration 1740
  - Broad View Of, In Enterprise Architect 1736

- Code Engineering
  - Code, Reverse Engineer 1790
  - Eclipse 1740
  - Element Context Menu 836
  - From Behavioral Models, Hardware Description Languages Supported 1778
  - From Behavioral Models, Software Languages Supported 1778
  - Generate Code For Single Class 1770
  - Generate Group of Classes 1771
  - Generate Package 1772
  - Generate Package Source Code 1772
  - Generate Source Code 1768
  - Hardware Description Languages Supported 1778
  - Introduction 1733
  - Languages Supported, Hardware Description Languages 1784
  - Namespaces 1777
  - Object Lifetimes 1898
  - Package Contents, Update 1774
  - Referenced XML Schema 2053
  - Reverse Engineer Source Code 1790
  - Settings 1891
  - Settings, Attribute/Operation Options 1899
  - Settings, Code Generation Constructor/Destructor Options 1898
  - Settings, Code Page for Source Editing 1900
  - Settings, General Code Options 1891
  - Settings, Import Component Types 1894
  - Settings, Source Code Options 1892
  - Supported Hardware Description Languages 1784
  - Supported Software Languages 1768
  - Synchronization 1790
  - Synchronize Model And Code 1776
  - Synchronize Package Tree 1774
  - UML Profile For XSD 2055
  - Unicode Character Set 1900
  - Update Package Contents 1774
  - Visual Studio 1740
  - With Enterprise Architect 1736
  - XML Schema 2027
  - XML Schema (XSD), Default UML To XSD Mappings 2048
  - XML Schema, Abstract XSD Models 2046
  - XML Schema, Generate XSD 2049
  - XML Schema, Import XSD 2053
  - XML Schema, Model XSD 2027
  - XSD 2027
  - XSD Datatype Packages 2065
- Code Engineering Submenu
  - Package Context Menu, Project Browser 575
- Code Generation
  - ActionScript Language Options 1909
  - Ada 2005 Language Options 1909
  - ArcGIS Language Options 1910
  - C Language Options 1911
  - C# Language Options 1912
  - C++ Language Options 1912
  - Delphi Language Options 1913
  - From Activity Diagrams 1778, 1787
  - From Behavioral Models 1778
  - From Behavioral Models, Hardware Description Languages Supported 1778
  - From Behavioral Models, Software Languages Supported 1778
  - From Interaction Diagrams 1778, 1786
  - From Sequence Diagrams 1778, 1786
  - From State Machine Diagrams 1778, 1780
  - Hardware Description Languages Supported 1778
  - Java Language Options 1915
  - Language Options 1907
  - Languages Supported, Hardware Description Languages 1784
  - Languages Supported, Options 1907
  - MDG Technology Language Options 1921
  - PHP Language Options 1916
  - Python Language Options 1917
  - Supported Hardware Description Languages 1778, 1784
  - Supported Languages 1741
  - Supported Languages, Options 1907
  - SystemC Language Options 1917
  - Template, Call From Other Template 1461
  - VB.NET Language Options 1918
  - Verilog Language Options 1919
  - VHDL Language Options 1919
  - Visual Basic Language Options 1920
- Code Generation Template
  - Transfer Between Models, Export 1400
  - Transfer Between Models, Import 1401
- Code Language
  - Create C# Automatic Properties 893
  - Create Properties As Attributes 892
- Code Language Options 1896
- Code Languages
  - Supported By Code Editor 1800
- Code Module
  - Add To MDG Technology 1331
- Code Sections
  - Synchronize 1403
- Code Template
  - Base Templates 1397



- Code Template
  - Custom Templates, Create 1462
  - Default Templates 1464
  - Editor 1404, 1462
  - Editor, Add New Stereotyped Templates 1464
  - Editor, Create Templates For Custom Languages 1466
  - Editor, In MDG Development 1462
  - Export 1462
  - Framework, In SDK 1394
  - Framework, Overview 1924
  - Import 1462
  - Overview 1395
  - Syntax, Introduction 1406
  - Syntax, Literal Text 1406
  - Syntax, Macros 1409
  - Syntax, Template Substitution Macros 1411
- Code Template Syntax
  - Variable Definitions 1407
  - Variable References 1407
  - Variables 1407
- Codepage
  - Set Up For RTF Report 2269
  - Set Up For RTF Report (Legacy) 2326
- Collaboration
  - Element 1184
  - Elements and Connectors, Now Communication 710
  - Message 1244
- Collaboration Diagram
  - Description 1084
  - Elements And Connectors 1084
  - Example 1085
  - Message Colors 1086
- Collaboration Use
  - Element 1186
- Collaborative Development 1900
- Collection Class
  - Additional 1912, 1915
  - Additional, Example Of Use 1906
  - Automation Interface Repository 2427
  - Example Of Use 1906
  - Set 1904
- Collections, EASL
  - Action 1451
  - Behaviors 1451
  - Classifier 1451
  - Construct 1451
  - Node 1451
  - State 1451
  - State Machine 1451
  - Transition 1451
- Trigger 1451
- Vertex 1451
- Color
  - Of Communication Messages 561
- Color Code External Requirements 1477
- Color Query
  - Shape Scripts 1372
- Column
  - Change Order 1944
  - Create In Data Modeling 1942
  - Data Modeling 1935
  - Definition 1942
  - In UML Data Modeling Profile 1942
  - Introduction 1935
  - Order, Change 1942
  - Properties 1942
  - Reorder 1944
  - Sequence Entries 1942
  - Stereotyped Attribute 1942
  - Unique 1942
- COM Interop
  - Debug .NET 1852
- COM Object
  - .NET Garbage Collect (Exit Method) 2448
- Combine
  - Windows In One Frame 135
- Combined Fragment
  - Create 1112
  - Element 1110
  - Interaction Operator 1113
- Comma Separated Value
  - Export 435, 439
  - Import 435, 440
- Command
  - Deploy, Create 1862
  - Run, Create 1862
  - Unit Test, Create 1834
  - Unit Test, Introduction 1834
- Commands
  - Add To Toolbar 154
  - Change Icon Appearance 154
  - Customize 154
  - Remove From Toolbar 154
- Commenting Selections
  - Code Editor, Common 1810
- Common
  - Connectors 706
  - Elements 706
  - Group, Toolbox 706
  - Relationships 706
- Communication
  - Connector 1218

- Communication
  - Elements and Connectors 710
  - Group, Toolbox 710
  - Message 1242
  - Message, Create 1243
  - Message, Level 1244
  - Message, Properties 1243, 1244
  - Message, Sequence 1244
  - Relationship 1218
- Communication Diagram
  - Description 1084
  - Elements And Connectors 1084
  - Example 1085
  - Labelled Associations 1084
  - Message Colors 1086
  - Numbering In 1084
- Communication Message
  - Colors 561
- Communication Path
  - Connector 1217
  - Relationship 1217
- Community Site 40, 87
- Compact
  - Project .EAP File 539
- Compare
  - Data 445
  - DDL With Database 1979
  - Diagram With Baseline 403
  - Models 445
  - Projects 445
  - Utility 394, 401
- Compare Utility
  - Context Menu 408
  - Keyboard Options 408
  - Merge Options 408
  - Options 402
  - Output 406
  - Tab 406
  - Toolbar 408
- Compartment
  - Constraint 842
  - Custom, Add To Elements Via Profile 1339
  - Element 842
  - Maintenance 842
  - Responsibility 842
  - Tag 842
  - Testing 842
- Compartment Events
  - Add-In Model 2608
  - EA\_GetCompartmentData 2609
  - EA\_QueryAvailableCompartments 2608
- Compiled November 8 2011 3
- Compiler Error
  - Build, Locate In Code 1866
- Complex Modeling
  - Enterprise Architect 9
- Component
  - Description 1023
  - Diagram 1023
  - Element 1187
  - Elements And Connectors 713, 1023
  - Example 1024
  - Group, Toolbox 713
  - Model Template 662, 670
  - View 676
- Compose
  - Connector 1217
  - Relationship 1217
- Composite
  - Elements And Connectors 709
  - Foreign Key 1959
  - Group, Toolbox 709
  - State 1152, 1153
  - State Regions 1037
- Composite Aggregation
  - Connector 1217
  - Relationship 1217
- Composite Diagram
  - Include Internal Embedded Elements 826
- Composite Element
  - Child Diagram 825
  - Drill Down 825
  - Icon 825
  - Include Internal Embedded Elements 826
  - Linked Diagram 825
  - Paste From Project Browser 736
- Composite Elements
  - Metaclass, Create With Supported Attributes 1314
- Composite Structure Diagram
  - Description 1017
  - Elements And Connectors 1017
  - Example 1018
- Compress
  - Timeline 1071
  - Transition 1071
- Computation Rule Table
  - Business Rule Modeling 1536
- Concept
  - ArchMate 1621
  - BPEL 1570
  - BPMN 1545
  - Eriksson-Penker MDG Technology 1623
  - Mind Mapping 1503

- Concept
  - SoaML 2088
  - SOMF 2092
  - SPEM 523
  - SysML 1990
- Concurrent Method Calls
  - In Add-Ins 2591
- Concurrent Substate
  - Regions 1037
- Conditional Node
  - Structured Activity 1161, 1168
- Conditional Substitution
  - Field Substitution Macros, Code Template Syntax 1413
- Configuration
  - Apache Tomcat Server 1849
  - JBOSS Server 1848
  - Tomcat Server 1849
  - Tomcat Service 1850
- Configure
  - Controlled Packages With XMI 426
  - Export To XMI Stubs 426
  - Local Options 542
  - Model Validation 2205
  - Options 542
  - Package For Version Control 362
  - Packages 426
  - User Registry Settings 542
- Configure Event Subtypes
  - Project Calendar 509
- Configure Timeline Dialog
  - States Tab 1062
  - Transitions Tab 1065
- Confirm
  - Parent Element 805
- Connect
  - Elements 943
  - Objects 943
  - To ASA Data Repository 219
  - To Automation Interface 2382
  - To Data Repository 209
  - To Element Feature 944
  - To MSDE Server Data Repository 221
  - To MySQL Data Repository 210
  - To Oracle 10g Data Repository, Via ODBC 214
  - To Oracle 10g Data Repository, Via OLE DB 216
  - To Oracle 11g Data Repository, Via ODBC 214
  - To Oracle 11g Data Repository, Via OLE DB 216
  - To Oracle 9i Data Repository, Via ODBC 214
  - To Oracle 9i Data Repository, Via OLE DB 216
  - To PostgreSQL Data Repository 218
  - To Progress OpenEdge Data Repository 221
  - To SQL Server Data Repository 212
- Connections
  - In Relationships Window 648
  - In Team Review 296
  - To Other Team Reviews 296
  - Window 648
- Connector 961
  - Activity Edge 1230
  - Add Between Locked Elements 273
  - Add Note 946
  - Add To Diagram 55
  - Add To UML Model, Quick Start 55
  - Add, Automation Interface Code Example 2573
  - Advanced Settings 968
  - Advanced, Menu Section 939
  - Aggregate 1208
  - Appearance 941
  - Appearance Options On Diagram 732
  - Application 714
  - ArcGIS 1634
  - Arrange 946
  - Assembly 1209
  - Associate 1210
  - Association 1210
  - Association Class 1215
  - Asynchronous Signal Message 1241
  - At Page Boundaries 948
  - Automation Interface, Connector Package 2508
  - Bend At Cursor 948
  - Bend Connector 55
  - Binding 964
  - Break Connection To Feature 944
  - Change Source Or Target 947
  - Change Type 947
  - Characteristics, Edit 939
  - Collaboration Numbers, Show 732
  - Communication 1218
  - Communication Path 1217
  - Compose 1217
  - Composite Aggregation 1217
  - Connect To Element Feature 944
  - Connector 1218
  - Constraints 963
  - Context Menu 936
  - Control Flow 1219
  - Copy Between Instances Of Elements From Project Browser 952
  - Create Between Elements 952
  - Create From Project Browser 952
  - Create From Toolbox 698

- Connector 961
  - Create In MDA-Style Transformation 1722, 1724
  - Create With Element Using Quick Linker 792
  - Create With Quick Linker 793
  - Create, Same Type As Previous 105
  - Crossed 948
  - Custom Properties, Set Value 939
  - Data Flow 1506
  - Delegate 1220
  - Delete 954
  - Dependency 1221
  - Dependency, Apply Stereotype 1221
  - Deployment 1222
  - Destination Role 967
  - Details 961
  - Direction Indicator 760
  - Display Options 559
  - Duplication In Transformation 1724
  - Edge 948
  - Entity Relationship Diagram 1627
  - Extend 1223
  - Extension (Profile Toolbox) 714
  - Field Substitution Macros 1423
  - Filter Display On Diagram 629, 631
  - Filter For Hidden Or Visible 631
  - Generalization 1224
  - Generalize 1224
  - Generalize (Profile Toolbox) 714
  - Hide 954
  - Hide Type On Diagram 629, 631
  - Hide/Show 941
  - Hide/Show On Diagram 732
  - Implements 1254
  - Include 1224
  - Information Flow 1225, 1226
  - Inheritance 1224
  - In-place Editor Options 957
  - Interrupt Flow 1230
  - Invoke 707
  - Jumps 948
  - Labels 760
  - Labels, Edit 957
  - Labels, Hide/Show On Diagram 732
  - Line Color 546
  - List For Requirements 1480
  - List, On Context References Tab 874
  - Locked 273
  - Manifest 1231
  - Message 1231
  - Move 57, 946
  - Multiplicity 964
  - Nesting 1249
  - Non-Navigable Ends, Show 732
  - Notation, Set 732
  - Notelink 1249
  - Notes 977
  - Object Flow 1250
  - Occurrence 1252
  - Off Page 948
  - Off-Page 936
  - Overview 1206
  - Package Import 1252
  - Package Merge 1253
  - Page Boundary 948
  - Pkg Import 1252
  - Pkg Merge 1253
  - Precede 707
  - Properties 961
  - Properties Menu 937
  - Properties In Transformation 1722
  - Realization, Common Diagram 1480
  - Realization, Quick Generation Of 1480
  - Realize 1254
  - Recursion 1255
  - Redefinition 714
  - Relationship 1218
  - Representation 1257
  - Represents 1256
  - Reverse Direction 958
  - Role Binding 1255
  - Role, Context Menu 936
  - Select By Property 629
  - Self-Message 1235
  - Shape Script Properties 1373
  - Source Role Properties 964
  - Stereotype Labels, Show On Diagram 732
  - Styles 940, 948
  - SysML Activity 1992
  - SysML Block Definition 1996
  - SysML Interaction 1999
  - SysML Internal Block 2000
  - SysML Model 2002
  - SysML Parametric 2005
  - SysML Requirement 2007
  - SysML State Machine 2009
  - SysML Use Case 2011
  - Tagged Value (Profile Toolbox) 714
  - Tagged Value, Use 1301
  - Tagged Values 967
  - Target Role Properties 967
  - Tasks 942
  - Template Binding 1257
  - To Class Created In Transformation 1724

- Connector 961
  - Trace 1260
  - Transform 1722, 1724
  - Transition 1260
  - Type Specific Menu Section 938
  - Type, Change 947
  - Types In Transformation 1722
  - Usage 1263
  - Use 1263
  - Visibility 941, 953, 955
  - What Is A? 1206
  - Working With 936
- Connector Package
  - Connector, Automation Interface 2508
  - ConnectorConstraint, Automation Interface 2514
  - ConnectorEnd, Automation Interface 2515
  - ConnectorTag, Automation Interface 2518
  - RoleTag, Automation Interface 2520
- Connector Package Diagram
  - Automation Interface 2507
- Connector Styles
  - Auto Routing 948
  - Bezier 948
  - Custom 948
  - Direct 948
  - Lateral 948
  - Line Style 948
  - Set 948
  - Tree Style 948
- ConnectorConstraint
  - Automation Interface, Connector Package 2514
- ConnectorEnd
  - Automation Interface, Connector Package 2515
- ConnectorTag
  - Automation Interface, Connector Package 2518
- Console
  - Commands, Scripting Window 2376
  - Tab, Scripting Window 2376
- Constant
  - Custom, For RTF Reports 2268
- ConstLayoutStyles Enum
  - Automation Interface 2414
- Constraint
  - Attach To Connector 793
  - Automation Interface, Element Package 2476
  - Compartment, Element 842
  - Composition 2195
  - Create 2195
  - Default, For SQL Server Table 1971
  - Define 2195
  - Delete 890
  - Element 850
  - Examples 2195
  - Expression 2195
  - Field Substitution Macros 1426
  - In Connector 963
  - In Scenarios & Requirements Window 879
  - Inherited, Show 745
  - Internal, Import As Test 2227
  - List 2195
  - Note, Element 1147
  - Of Attributes, Create 890
  - Operand 2195
  - Post Condition On Actions 1093
  - Precondition On Actions 1093
  - Profile 1302
  - Resequence 2195
  - Status Type, Define 998
  - Stereotype 1302
  - Synchronize, And Tagged Values 1281
  - Synchronize, From MDG Toolbox Pages 1281
  - Tab, Scenario 875
  - Type, Define 997
- Contents
  - Table, MS Word, Add 2348
- Contents Submenu
  - Package Context Menu, Project Browser 577
- Context Diagram 1506
- Context Element
  - Highlight 822
  - In Multiple Selection 822
- Context Item Events
  - Add-In Model 2611
  - EA\_OnContextItemChanged 2612, 2614
  - EA\_OnContextItemDoubleClicked 2613
- Context Menu
  - Apply «Stereotype» Option 698, 1281
  - Attribute, Project Browser 582
  - Code Editor, Common 1805
  - Configure Code Editor Search Options 1805
  - Connector 936
  - Connector Label 760
  - Connector Role 936
  - Diagram 687
  - Diagram, Project Browser 581
  - Element 827
  - Element Label 760
  - Element, Add Supporting Diagrams and Elements 832
  - Element, Multiple Selection 840
  - Element, Project Browser 577
  - Item, Structural Specification 859
  - Linked Document Editor 930

- Context Menu
  - Main 90
  - Method, Project Browser 582
  - Model Views 601
  - Model, Project Browser 569
  - New Element Or Connector 690
  - Operation, Project Browser 582
  - Package, Project Browser 571
  - Project Browser 569
  - Record & Analyze 2151
  - Scripting Window 2373
  - Selected Text, Structural Specification 861
  - Structural Specification Entry Points 863
  - Team Review Options 280
- Context Reference
  - Add Comments, On Context References Tab 874
  - Add, On Context References Tab 874
  - Delete, On Context References Tab 874
  - List, On Context References Tab 874
  - Tab 874
- Continuation
  - Element 1155
- Control
  - Create 1665
  - Element 1665
- Control Flow
  - Connector 1219
  - Guard 1219
  - Relationship 1219
  - Weight 1219
- Control Macros
  - Code Template Syntax 1444
- Control Recording
  - Execution Analysis, Recording Sequence Diagrams 2161
- Control Stack Depth 2153
- Controlled Package 362
  - Batch Export To XMI 430
  - Batch Import From XMI 431
  - Disconnect 428
  - Load 429
  - Menu, XMI 424
  - Recovery 432
  - Report Deletion Of Cross Package References In XMI 433
  - Version Control 432
  - With XMI 422
- Converge Diagram Layout 784
- Convert
  - Linked Element To Local Copy 823
  - Names In MDA Transformations 1728
- CONVERT\_DB\_TYPE 1728
- CONVERT\_NAMES
  - Macro 1728
- CONVERT\_TYPE 1728
- Convey
  - Information Item 1228
- Copy
  - Attributes Between Elements 823
  - Base Project 190
  - Diagram Image To Clipboard 742
  - Diagram Image To Disk File 741
  - Diagram, Deep 742
  - Diagram, Shallow 742
  - Element Between Packages 808
  - Element On Diagram 737
  - Elements Between Diagrams 57
  - Existing Project 191
  - Maintenance Item Between Categories 2240
  - Model 183, 185, 187
  - Operations Between Elements 823
  - Package 683
  - Packages Between Projects 444
  - RTF Bookmark To Clipboard 575, 581
  - Test Between Categories 2223
  - UML Diagram, Deep 742
  - UML Diagram, Shallow 742
- Copyright Notice 27
- CORBA
  - MDG Technology For, Enterprise Architect 1289
- Co-Region Notation 1242
- Corporate Edition
  - Of Enterprise Architect 22
- Correct Spell Checked Words 486
- Corrupt EAP file 540
- Create
  - A Project In Enterprise Architect 49
  - Access 2007 Repository 194
  - Activity, BPEL 1.1 1581
  - Activity, BPEL 2.0 1605
  - Adaptive Server Anywhere Repository 198
  - Add-In 2587
  - Attribute Properties 892
  - Automatic Properties, C# 893
  - Baselines 400
  - Boundary Element 1664
  - BPEL 1.1 Assignments 1589
  - BPEL 1.1 Model 1573
  - BPEL 1.1 Web Service 1591
  - BPEL 2.0 Model 1597
  - BPEL 2.0 Web Service Operation 1615
  - BPEL 2.0 WSO 1615

- 
- Create
- Build Script 1832
  - Columns In Data Modeling 1942
  - Combined Fragment 1112
  - Communication Messages 1243
  - Composite Foreign Key 1959
  - Connector With Quick Linker 793
  - Control Element 1665
  - Custom Diagram Background, Image Manager 757
  - Custom Tagged Values 1393
  - Custom View, Add-In Model 2670
  - Data Object, BPEL 2.0 1610
  - Data Repository 194
  - Design Master, Replication 246
  - Diagram 724
  - Diagram From Linked Document 932
  - Diagram, Automatically 681
  - Document Artifact 929
  - Documents 2249
  - Element And Connector With Quick Linker 792
  - Element From Linked Document 932
  - Element From Maintenance Item 2241
  - Element From Text File 797
  - Element From Text File Text 1473
  - Element In Diagram 797
  - Element On Diagram From Project Browser 733
  - Element Template 820
  - Elements 797
  - End Event, BPEL 1.1 1586
  - End Event, BPEL 2.0 1608
  - Entity 1667
  - Favorites Folder, Model View 600
  - Favorites Folder, Model View (Context Menu) 601
  - Foreign Key 1956
  - Gateway, BPEL 1.1 1584
  - Gateway, BPEL 2.0 1607
  - Hidden Submenu In Toolbox Profile 1341
  - HTML Report 2353
  - Intermediate Event, BPEL 1.1 1578
  - Intermediate Event, BPEL 2.0 1602
  - Learning Center Profiles 1350
  - Link Between Elements 952
  - Linked Document Template 934
  - Masked Tagged Values 1393
  - MDG Technologies, Overview 1324
  - Model 189
  - MSDE Server Repository 198
  - MySQL Repository 195
  - Notes 814
  - Oracle 10g Server Repository 197
  - Oracle 11g Server Repository 197
  - Oracle 9i Server Repository 197
  - Pattern 1271
  - Pool, BPEL 1.1 1588
  - Post In Team Review 287
  - PostgreSQL Repository 197
  - Primary Key 1952, 1953
  - Primary Key Name Template 1952
  - Profiles 1296
  - Progress OpenEdge Repository 199
  - Project 189
  - Project File 178
  - Property, BPEL 2.0 1611
  - Reference Data Tagged Values 1392
  - Relationship Using Matrix 644
  - Replicas 246
  - Requirement From Text File Text 1473
  - Requirements 1473
  - Rich Text Format Report (Enhanced Generator) 2253
  - Root Node, Model View 600
  - Root Node, Model View (Context Menu) 601
  - RTF Report (Enhanced Generator) 2253
  - RTF Style Template (Legacy) 2323
  - RTF Template 2270
  - Search Definition 623
  - Sequence Flow, BPEL 1.1 1587
  - Sequence Flow, BPEL 2.0 1612
  - Slideshow, Model View 600
  - Slideshow, Model Views 607
  - SQL Server Repository 195
  - Start Event, BPEL 1.1 1577
  - Start Event, BPEL 2.0 1601
  - Structured Tagged Values 1389
  - Table in Data Modeling 1936
  - Team Review Category 284
  - Team Review Topic 285
  - Technology Files (Option) 115
  - Text 814
  - Timing Diagram 1053
  - Timing Message 1246
  - Toolbox Profile For MDG Technology 1337
  - UML Diagram 724
  - UML Pattern 1271
  - Views Folder, Model View 600
  - Views Folder, Model View (Context Menu) 601
  - Views, Model View 600
  - Views, Model View (Context Menu) 601
  - Create A Breakpoint Or Recording Marker Option 1805
  - Create Property Implementation Dialog 892
-

- CreateBaselineFlag Enum
    - Automation Interface 2416
  - CreateModelType Enum
    - Automation Interface 2416
  - CreateObjectAction 1787
  - CreateOperationAction 1787
  - Cross Package References
    - XMI, Report Deletion 433
  - Cross Reference
    - Between EMX Files 419
    - Delete 803
    - Set For Element 803
    - Trace With Traceability Window 637
    - Use In Element 803
  - Crossed
    - Connectors 948
  - CSV
    - Export 435, 439
    - Export From Relationship Matrix 645
    - Export From Rule Composer 1536
    - Export State Machine Table To 1048
    - Import 435, 440
    - Import Requirement Hierarchies 1481
    - Import Requirements 1481
    - Preserve Hierarchy 438
    - Specifications 435
  - CTF
    - In SDK 1394
    - Overview 1924
  - Current Connector
    - Toolbar 148
  - Current Element
    - Toolbar 146
  - Cursor History
    - Code Editor, Common 1810
  - Custom
    - Diagram 1494, 1655, 2234
    - Diagram Types 1346
    - Elements and Connectors 716
    - Group, Toolbox 716
    - Stereotypes 1293
  - Custom Background
    - Create For Diagram, Image Manager 757
  - Custom Compartment
    - Add To Elements Via Profile 1339
  - Custom Diagram
    - Description 1505
    - Elements And Connectors 1505
    - Model 1505
  - Custom Language
    - Create Templates For In Code Template Editor 1466
  - Settings, RTF Report Generation 2269
  - Custom Reference
    - Delete 803
    - Set For Element 803
    - Trace With Traceability Window 637
    - Use In Element 803
  - Custom Tagged Values
    - Create 1393
  - Custom Template
    - Create From Project 189
  - Custom Tools
    - External Applications 157
  - Custom View
    - Add-In Model 2669
  - Customize
    - Commands 153, 154
    - Diagram Appearance 725
    - Dialog 153
    - External Tools, Pass Parameters To 159
    - Keyboard 153
    - Keyboard Shortcuts 160
    - Menu Appearance 161
    - Menus 153
    - Model Search 619
    - Options 153
    - RTF Language 2269
    - RTF Language (Legacy) 2326
    - Toolbar Option Appearance 162
    - Toolbars 153, 155
    - Tools 153, 156
    - Visibility Of Elements 813
    - Window 153
  - CustomProperties Collection
    - Automation Interface, ElementFeatures Package 2399
  - Cut And Paste
    - Elements Between Diagrams 57
  - CVS
    - Package Version Control History, In 376
    - Version Control Options 350
- D -
- Dashed Border
    - On Element 805
  - Data
    - Breakpoint, Set 1882
    - Compare 445
    - Export 306
    - Import 308
    - Integrity 535
    - Integrity Check 535



- Data
  - Integrity, Run SQL Patches 539
  - Model Template 662, 669
- Data Breakpoint
  - Set 1882
- Data Distribution Service
  - MDG Technology For, Enterprise Architect 1289
- Data Flow
  - Concepts 1506
  - Connector 1506
  - Context Diagram 1506
  - Diagram 1506
  - MDG Technology 1506
  - Relationship 1506
  - Toolbox Page 1506
- Data Management
  - Project Compare 119
  - Project Integrity 119
  - Project Transfer 119
  - Submenu (Tools Menu) 119
- Data Model
  - Physical 1932
  - To ERD Transformation, MDA-Style Transform 1689
  - Transformation From Entity Relationship Diagram 1697
  - Transformation To Entity Relationship Diagram 1689
- Data Modeling
  - Change Table Owner 1945
  - Check Constraint 1966
  - Check Constraint, Create 1966
  - Compare DDL With Database 1979
  - Create Columns 1942
  - Create Data Model Diagram 1933
  - Create Table 1936
  - Data Model Diagram 1934
  - Data Type Conversion, Table 1946
  - Data Types, Introduction 1944
  - Database View 1967
  - Database View, Create 1968
  - DBMS Conversion Procedure, Package 1945
  - DBMS Data Types 1947
  - DDL, Generate 1978
  - Elements And Connectors 720
  - Foreign Keys 1956
  - Generate DDL 1978
  - Generate DDL For A Package 1979
  - Group, Toolbox 720
  - Index 1963
  - Index, Create 1963
  - Introduction 1625
  - Notation 1982
  - Primary Key Extended Properties 1955
  - Primary Key, Create 1952
  - Profile (UML) 1625
  - Set Database Type 1938
  - Set MySQL Table Type 1939
  - Set Oracle Table Properties 1940
  - Set Schema Owner 1939
  - Set Table Owner 1939
  - Set Table Properties 1937
  - Stored Procedure 1961
  - Tables And Columns 1935
  - Trigger 1965
  - Trigger, Create 1965
  - Typical Tasks 1625
- Data Object
  - Create In BPEL 2.0 1610
  - Model In BPEL 2.0 1610
- Data Repository
  - Adaptive Server Anywhere, Connect To 219
  - Connect To 178, 209
  - Create 194
  - MSDE Server, Connect To 221
  - MySQL, Connect To 210
  - Oracle 10g, Connect To, Via ODBC 214
  - Oracle 10g, Connect To, Via OLE DB 216
  - Oracle 11g, Connect To, Via ODBC 214
  - Oracle 11g, Connect To, Via OLE DB 216
  - Oracle 9i, Connect To, Via ODBC 214
  - Oracle 9i, Connect To, Via OLE DB 216
  - PostgreSQL, Connect To 218
  - Progress OpenEdge, Connect To 221
  - SQL Server, Connect To 212
- Data Source
  - Select 1975
- Data Store
  - Element (Data Flow Diagram) 1506
- Data Transfer
  - Between Repositories 442
  - Compare Projects 445
  - Copy Packages Between Projects 444
  - Transfer Project Data 442
  - Upsize to MySQL 224
- Data Type
  - Add 987, 1947
  - Code 987
  - Conversion, Table 1946
  - Data Modeling, Introduction 1944
  - Definition 422
  - Delete 987, 1947
  - Element 1188

- Data Type
  - Extend 987
  - For MySQL 1949
  - Instance 1188
  - Map Between DBMSs 1947, 1950
  - Modify 987
  - Programming Language 987
  - Referenced 1188
  - Set For DBMS 1947
- Database
  - Compare Package DDL With 1979
  - Default 1895
  - Design 1625
  - Keys 1625
  - Model Template 662, 669
  - Modeling 1625
  - Schema, Import Of 1625
  - Set Default 142
  - Supported Types 1625, 1970
  - View 1968
  - View, What Is A 1967
- Database Administrator
  - And Enterprise Architect 84
  - Project Role 84
- Database Engineering
  - Introduction 1931
  - Physical Data Model 1932
  - Submenu (Tools Menu) 117
- Database Keys 1951
- Database Modeling
  - Enterprise Architect 9
- Database Object
  - Select From ODBC Data Source 1976
- Database Operation Properties
  - Dialog 1955
- Database Repository
  - Access Permissions For 193
  - Connect To 193
  - Create 193
  - Set Up 193
- Database Schema
  - Description 1631
  - Diagram 1631
  - Elements And Connectors 1631
  - Example 1631
- Database Table
  - Select From ODBC Data Source 1976
- Database Type
  - Set For Table 1938
- Database View
  - Create 1968
  - Definition 1968
- Example 1968
- Report On 1968
- Datastore
  - Element, Activity Diagram 1116
- Datatype
  - Automation Interface Repository 2429
- Datatypes
  - For Oracle 1949
- DBMS
  - Conversion, For A Table 1946
  - Data Type Conversion, Table 1946
  - Map Data Types Between 1950
  - Set As Default For Model 1947
  - Set Datatypes For 1947
  - Supported Types 1970
- DBMS Conversion
  - Mapper 1945
  - Procedure 1945
  - Table Conversion Between DBMS Types 1945
  - Table Owner, Change 1945
- DDL
  - Compare With Database 1979
  - Data Modeling 1978
  - Default Script Editor 1895
  - Generate 1978
  - Generate For Package 1979
  - Generate For Table 1978
  - Import Schema From ODBC 1973
  - Schema, Import From ODBC 1973
  - Scripts And Generated Tables 1625
  - Transformation 1689
- DDS
  - MDG Technology For, Enterprise Architect 1289
- Debug
  - .NET 1837, 1850
  - .NET CLR Versions 1851
  - .NET With COM Interop Process 1852
  - Another .NET Process 1888
  - Application, Analyzer menu Option 120
  - ASP .NET 1853
  - Break On Variable Changing Value 1882
  - C, General Setup 1841
  - C++, General Setup 1841
  - Code 1824
  - Create Sequence Diagram, Call Stack 1886
  - Deploy Script, Create New 1862
  - Enterprise Architect 9
  - File Search, Use 1817
  - Inspect Process Memory 1886
  - Intermediary Language, MDA Transformation 1714

- Debug
  - Java 1837, 1843
  - Java Applets In Internet Browsers 1845
  - Java Web Servers 1846, 1850
  - Java, Advanced Techniques 1845
  - Java, General Setup 1843
  - Microsoft Native Applications 1841
  - On Windows 7 And Windows Vista 1838
  - Platforms 1837
  - Process First Chance Exceptions 1889
  - Run Script, Create New 1862
  - Script Search 1817
  - Search Window 1817
  - Show Loaded Modules 1887
  - Show Output 1875
  - Step Into Function Calls 1875
  - Step Out Of Functions 1875
  - Step Over Lines Of Code 1875
  - Step Through Function Calls 2162
  - Submenu (Analyzer) 122
  - Tooltips In Code Editor 1879
  - Tracepoint Statement Output 1860
  - Under Windows Vista 1837
  - Unit Test Script, Create 1834
  - Unit Test Script, Introduction 1834
  - View Call Stack 1884
  - View Local Variables 1877
  - View Local Variables, Long Values 1879
  - View Variables In Other Scopes 1880
  - Visual Basic, General Setup 1841
  - WINE Applications 1839
- Debug & Record
  - Toolbar 150
- Debug Session
  - Debug C++ 1841
  - Java, Attach To VM 1845
  - Microsoft Native Setup 1841
  - Set Up 1837
  - Set Up For .NET 1850
  - Set Up For Microsoft Native 1841
- Debug Symbols
  - Debug C++ 1842
  - Microsoft Native 1842
- Debugger 1861
  - Actions 1867
  - Before Starting 1867
  - Compare Variable Snapshots 1881
  - Debug Another Process 1888
  - Detach From Process 1888
  - Frameworks 1867
  - GDB 1859
  - GNU 1859
  - Introduction 1867
  - Modules Window 1887
  - On Windows 7 And Windows Vista 1838
  - Overview 1867
  - PHP, Checklist 1857
  - PHP, Scenarios 1854
  - PHP, System Requirements 1857
  - PHP, Timeouts 1857
  - Process 1867
  - Process First Chance Exceptions 1889
  - Start 1875
  - Stop 1875
  - System Requirements 1837
  - Toolbar 150
  - View Elements Of Array 1881
- Debugging Actions 1874
- Decision
  - Element 1117
- Decision Table
  - Business Rule Modeling 1536
  - Rule Action Section 1536
  - Rule Binding Section 1536
  - Rule Condition Section 1536
- Deep Copy
  - Of Diagram 742
- Deep History 1134
  - Change From Shallow 829
- Default
  - Code Language, Set 142
  - Database 1895
  - Database, Set 142
  - DDL Script Editor 1895
  - Hours 520
  - Model Diagram, Cancel (Menu Option) 106
  - Model Diagram, Set (Menu Option) 106
  - Project Browser Behavior 587
  - Templates 1464
  - UML To XSD Mappings 2048
  - User Diagram, Cancel (Menu Option) 106
  - User Diagram, Set (Menu Option) 106
- Default Appearance
  - Background Color 817
  - Border Color 817
  - Border Thickness 817
  - Font Color 817
  - Of An Element 837
  - Of Element 817
  - Set For Profile Stereotype Objects 1306
- Default Constraint 1971
- Default Diagram
  - Model, Cancel (Menu Option) 106
  - Model, Cancel For 744

- Default Diagram
  - Model, Set (Menu Option) 106
  - Model, Set For 744
  - User, Cancel (Menu Option) 106
  - User, Set (Menu Option) 106
- Default Fonts
  - Model 552
  - Set 552
  - User 552
- Default Hours
  - Estimation 520
  - Per Adjusted Use Case Point 520
  - Project Management 520
  - Rate 520
  - Settings 520
- Default Settings
  - Options Dialog 153
- Default Templates
  - Override in Code Template Editor 1464
- Default Toolbox
  - Override In Profile 1342
- Default Tools Toolbar 139
- Defect
  - Add 2244
  - Create From Test Item 2228
  - Element 2244
  - Hide Stereotype Letter 2244
  - Show Stereotype Letter 2244
- Define
  - Author 989
  - Browser Behavior 544
  - Clients 993
  - Clipboard Image File Format 544
  - Email Exchange Server 544
  - File Directory 544
  - Foreign Key Name Template 1960
  - Home Web Site 544
  - Internet Search Engine 544
  - Learning Center Profile Toolbox 1350
  - Resources 992
  - Roles 991
  - Run-Time Variable 1198
  - Stereotype As Metatype 1312
  - Stereotype Constraints 1302
  - Testpoint Constraint 2195
  - Validation Configuration For MDG Technology 1354
- Define Menu Items
  - Create Add-In 2588
- Defined Environment Types 518
- Delegate
  - Connector 1220
  - Relationship 1220
- Delete
  - Category In Team Review 291
  - Connectors 954
  - Connectors, Quick Start 62
  - Diagram (Single) From Project Browser 581
  - Diagrams (Multiple) From Project Browser 581
  - Diagrams From Project browser 738
  - Diagrams, Quick Start 62
  - Element Changes 2238
  - Element Defects 2238
  - Element From Diagram 811
  - Element from Model 811
  - Element From Project Browser 811
  - Element Issues 2238
  - Element Tasks 2238
  - Elements, Impact of Auditing 393
  - Elements, Quick Start 62
  - Instance Variable 1198
  - Item In Team Review 291
  - Line Points 948
  - Linked Document 933
  - Linked Document Template 934
  - Locks 268
  - Menu Options 161
  - Package Attributes From Model Document 2334
  - Package In Project Browser 685
  - Packages From Model Document 2334
  - Packages, Quick Start 62
  - Post In Team Review 291
  - Project Task 461
  - Relationship 851
  - Relationship Using Matrix 644
  - Reply In Team Review 291
  - RTF Template 2270
  - Team Review Category 280
  - Team Review Post 280
  - Team Review Resource 280
  - Team Review Topic 280
  - Topic In Team Review 291
  - Views 678
- Delete Selected Element(s)
  - Menu Option (Edit Menu) 93
- Deletion
  - And Auditing 393
- Delphi
  - Code Generation 1913
  - Import, Reverse Engineering 1793
  - Language Options 1913
  - Limitations 1914
  - Modeling Conventions 1754

- Delphi
  - Properties 1914
- Demonstration
  - Of Enterprise Architect 86
- Dependency
  - Connector 1221
  - Relationship 1221
  - Relationship, Apply Stereotype 1221
  - Report, Generate 2339
- Dependency Report
  - In Traceability 636
- Deploy
  - Add-In 2589
  - Application, Analyzer menu Option 120
  - MDG Technology From Add-In 1357
  - MDG Technology From File 1357
- Deploy Command
  - Create 1862
- Deploy Script
  - Create 1862
- Deployment
  - Connector 1222
  - Diagram 1020
  - Elements and Connectors 713
  - Group, Toolbox 713
  - Model Template 662, 671
  - Relationship 1222
  - View 676
- Deployment and Rollout
  - And Enterprise Architect 81
  - Project Role 81
- Deployment Diagram
  - Description 1020
  - Elements And Connectors 1020
  - Example 1021
- Deployment Spec
  - Element 1189
- Derived
  - Attribute 887
  - Symbol 887
- Design
  - Patterns 1271
- Design Master 244, 538
  - Create, Replication 246
- Design Systems Using UML
  - Enterprise Architect 9
- Designate Driving Triggers
  - Model State Machine For HDL 1784
- Desktop Edition
  - Of Enterprise Architect 22
  - Upsize From 193
- Desktop Tools
  - Add 156
  - Configure 156
  - Customize 156
- Destination Role 967
- DestroyObjectAction 1787
- Developer
  - And Enterprise Architect 78
  - Forward Engineering 78
  - Project Role 78
  - Reverse Engineering 78
  - Round-Trip Engineering 78
  - Visualise Package Arrangement 78
- Device
  - Element 1190
- Diagram 727
  - Activity, Description 1028
  - Activity, Generate From Scenarios 865
  - Add And Manage, Automation Interface Code Example 2574
  - Add Elements Via Context Menu 690
  - Add Link To Team Review Post 292
  - Add Profile Feature 1280
  - Add Profile Object 1280
  - Add To Project 724
  - Add To UML Model, Quick Start 52
  - Advanced Menu 106
  - Alias 727
  - Alternative Image For Element 755
  - Analysis 1509
  - Anti-Aliased Rendering 550
  - Anti-Aliased Text 550
  - Appearance Options 550
  - Appearance Options, Connectors 732
  - Appearance Options, Diagram Tab 727
  - Appearance Options, Element 729
  - Appearance Options, Features 731
  - Appearance Options, General 726
  - Appearance Options, Set 725
  - Appearance Options, Visible Class Members 732
  - ArchiMate 1621
  - Attribute Details, Show 731
  - Auto Route Layout 786
  - Automatic Layout 787
  - Automatic Save 553
  - Automation Interface, Diagram Package 2523
  - Background Color 546
  - Background Color Gradient 550
  - Behavior Options 553
  - Behavioral, Overview 1027
  - Box Layout 775
  - BPEL 1570

- Diagram 727
  - BPMN 1545
  - Build 691
  - Business Interaction 1514
  - Business Modeling 1514
  - Business Process 665
  - Cancel Model Default 744
  - Center-Focussed Layout 770
  - Change Type 740
  - Change Type (Menu Option) 106
  - Circular Layout 770
  - Class 668, 1014
  - Class Features, Visibility 549
  - Close 691
  - Collaboration 1084
  - Communication 1084
  - Compare With Baseline 403
  - Component 670, 1023
  - Composite Structure 1017
  - Connector Appearance Options 732
  - Connector Notation, Show 732
  - Context 1506
  - Context Menu 687
  - Context Menu, Project Browser 581
  - Converge Layout 784
  - Copy Image 105
  - Copy, Deep 742
  - Copy, Shallow 742
  - Create 724
  - Create Automatically 681
  - Create Custom Background, Image Manager 757
  - Create From Linked Document 932
  - Create Using Image Library 758
  - Creator 726
  - Custom 1494, 1505, 1655, 2234
  - Customize Appearance 725
  - Data Flow 1506
  - Data Model, Create 1933
  - Data Model, Example 1934
  - Database Schema 669, 1631
  - Define Child Type, Supported Attributes 1315
  - Delete (Multiple) From Project Browser 738
  - Delete (Single) From Project Browser 738
  - Delete Element From 811
  - Delete From Project Browser 581
  - Delete Multiple Elements From 811
  - Deployment 1020
  - Details Note 727, 747
  - Diagram 1508, 1665, 1666
  - Diagram To Package 52
  - Digraph Layout 778
  - Display Options 549
  - Diverge Layout 784
  - Divide Between Pages In RTF Reprt 727
  - Drag & Drop Elements From Project Browser 733
  - Drag Existing Package Onto 684
  - Duplicate 742
  - Element Appearance Options 729
  - Element Compartments, Show/Hide 729
  - Element Icons 841
  - Element Stereotypes, Show 729
  - Element, Copy 737
  - Element, Paste 737
  - Elliptical Layout 770
  - Entity Relationship 1627
  - Eriksson-Penker 1623
  - Exclusive Edit Lock 253
  - Extended 1010
  - Extended UML 1502
  - Fan Relations Layout 783
  - Feature Return Types, Show 731
  - Feature Stereotypes, Show 731
  - Features Appearance Options 731
  - Filter Display Of Connectors 629, 631
  - Filter Display Of Elements 629, 631
  - Find In Project Browser 105
  - Find Related Elements 738
  - Frame 549, 1119
  - Frame (Border) 1119
  - Generate From Scenarios 864
  - Grid Size, Adjust 553
  - Hand Drawn Mode 727
  - Hyperlink 1119
  - Hyperlink To 1670
  - Increase Display Size 550, 687
  - Interaction 1051, 1074, 1084, 1087
  - Interaction Overview 1087
  - Layout Options 787
  - Layout Tool 769
  - Layout, Move Sections 763
  - Legend 748
  - Lock, General (Menu Option) 106
  - Lock, Require User Lock 274
  - Lock, Security Off 767
  - Lock, User/Group Lock 271
  - Logical 1014
  - Maintenance 2234
  - Make All Elements Selectable 687
  - Make Model Default 744
  - Manage Display 691
  - Managing 721
  - MDG Technology 1010

- Diagram 727
- Menu 105
  - Mind Mapping 1503
  - Modeling With 687
  - Modify Z-Order, Of Elements 687
  - Move Elements 805
  - Move Sections 763
  - Move, Impact On Element 57, 1117, 1127, 1128, 1136
  - Navigation And Selection Hotkeys 740
  - Neaten Layout 781
  - Note 747
  - Notes 726
  - Notes, Show/Hide 549
  - Object 1015
  - Open From Shortcut 183, 187
  - Open From Shortcut (Direct Definition) 185
  - Open Package In 744
  - Overview 1010
  - OWL Definition 1651
  - OWL Facts 1651
  - Package 1011
  - Page Setup 549
  - Pan And Zoom 610
  - Parametric, SysML 2013
  - Paste 742
  - Per Page Layout 776
  - Place Related Elements On Current 738
  - Print From Project Browser 581
  - Print Page Footer 727
  - Print Page Header 727
  - Profile Attributes, PData and StyleEx 1348
  - Profiles 1346
  - Properties (Diagram Menu) 105
  - Properties Dialog - Connectors Tab 732
  - Properties Dialog - Diagram Tab 727
  - Properties Dialog - Elements Tab 729
  - Properties Dialog - Features Tab 731
  - Properties Dialog - General Tab 726
  - Properties Dialog, Visible Class Members, 732
  - Properties Note, Add 747
  - Properties, Set 725
  - Property Strings, Show 732
  - Qualifiers, Show 731
  - RDF Definition 1651
  - Redo Last Action 768
  - Reference 1119
  - Relationship Traceability For 637
  - Relationships, Show 732
  - Rename 739
  - Requirements 666, 1494
  - Robustness 1074, 1084, 1508, 1509, 1663, 1665, 1666
  - Robustness, Generate From Scenarios 871
  - Rotate Image In RTF Report 727
  - RTF Document Options 727, 2255
  - RTF Report On Elements Linked From Other Packages 2255
  - Rule Model 1527
  - RuleFlow, Generate From Scenarios 867
  - Save 105, 687
  - Save As Pattern 1271
  - Save As UML Pattern (Menu Option) 106
  - Save Automatically 553
  - Save Image Of 105
  - Save Profile 1307
  - Scale View 550
  - Schema Diagram 1631
  - Scripts 2375
  - Scroll Through From Diagram Toolbar 145
  - Sequence 1074
  - Sequence, Generate From Scenarios 869
  - Set Page Size 764
  - Show As Diagram List 687
  - Show Diagram List As 592
  - Show Realised Interfaces For Class 760
  - Show/Hide Package Contents 685
  - Slideshow, Model View 598, 607
  - SoaML 2088
  - SOMF 2092
  - SPEM 524
  - Spring Layout 780
  - Stae Machine, Generate From Scenarios 867
  - State 1032
  - State Machine 1032
  - Stereotype 726
  - Structural, Overview 1011
  - Swimlanes 751
  - Swimlanes Matrix 753
  - Switch Between 691
  - Synchronize With Scenario Steps 864
  - SysML 1990
  - SysML, Parametric 2013
  - Tab Context Menu 695
  - Table Owner, Show 729
  - Tabs, Close 695
  - Tabs, Switch Between 697
  - Tasks, General 721
  - Test Domain 2196
  - Timing 1051
  - Toolbar 145
  - Traceability 649
  - Types 1010

- Diagram 727
  - Types, Built In 1348
  - Types, Custom 1346
  - UML 1010
  - Undo Last Action 768
  - Use Case 666, 1030
  - User Interface 1655
  - Version 726
  - View 691
  - View Next 764
  - View Previous 764
  - Visibility Indicators 731
  - Visible Relations (Menu Option) 106
  - Visual Check For Changes 403
  - Water Mark 550
  - What Is A? 1010
  - Whiteboard Mode 727
  - Working With 687
  - WSDL Binding 2078
  - WSDL Message 2073
  - WSDL Overview 2070
  - WSDL Port Type 2075
  - WSDL Service 2082
  - WSDL Types 2067, 2070
  - Zoom From Diagram Toolbar 145
  - Z-Order Element 744
- Diagram Caption Bar
  - Hide 98
  - Show 98
- Diagram Filters
  - Access 631
  - Application 629
  - Clear Effect Of 631
  - Create 631
  - Delete 631
  - Disable 631
  - Enable 631
  - Introduction 629
  - Set Effect Of 631
  - Suggested Use 629
  - Window 631
- Diagram Frame 1119
- Diagram Gate
  - Element 1120
- Diagram Image
  - Copy To Clipboard 742
  - Copy To Disk File 741
  - Save To Disk File 741
- Diagram List
  - Context Menu Options 592
  - Description 596
  - Filter Bar 590
- Generate RTF Report 2253
- Options 592
- Show As Diagram 592
- Show Diagram As 687
- Toolbar 592
- Value Grouping 590
- View Header 590
- Work On Elements 592
- Diagram Note
  - Insert New From Toolbar 143
- Diagram Only Report
  - Generate 2340
- Diagram Package
  - Automation Interface 2522
  - Diagram, Automation Interface 2523
  - DiagramLinks, Automation Interface 2528
  - DiagramObject, Automation Interface 2529
  - Swimlane, Automation Interface 2534
  - SwimlaneDef, Automation Interface 2531, 2532
- Diagram Tabs
  - Switch Between 697
- Diagram Toolbox 698
  - Activity Group 712
  - Analysis Group 715
  - ArcGIS 1634
  - Class Group 708
  - Common Group 706
  - Communication Group 710
  - Component Group 713
  - Composite Group 709
  - Custom Group 716
  - Data Modeling Group 720
  - Deployment Group 713
  - Interaction Group 710
  - Maintenance Group 717
  - Metamodel Group 715
  - Object Group 709
  - ODM Pages 1644
  - Profile Group 714
  - Requirement Group 717
  - Shortcut Menu 704
  - State (Machine) Group 711
  - Test Domain Group 721
  - Timing Group 711
  - Use Case Group 707
  - User Interface Group 718
  - WSDL Group 719
  - XML Schema Group 720
- Diagram Type
  - Add To MDG Technology 1329
- DiagramLinks
  - Automation Interface, Diagram Package 2528



- DiagramObject
  - Automation Interface, Diagram Package 2529
- Dialog
  - All User Permissions 261
  - Application Look 162
  - Attach To Process 1888
  - Audit Settings 384
  - Baseline Diagram Compare 403
  - Class Invariant 2193
  - Class Markup Selection 2160
  - Create Property Implementation 892
  - Database Operation Properties 1955
  - Difference 387
  - Find And Replace 1814
  - General Types 995
  - Group Properties (Script) 2375
  - Link Note To Element Feature 815
  - Maintenance 1001
  - Manage Uses Context List 861
  - Model Message 492
  - New Action 1095
  - Operation Constraints 2193
  - Options 542
  - Package Control Options 426
  - Paste Element 733
  - People 989
  - Project Issues 462
  - Qualifiers 1212
  - Recent Post Options 601
  - Select <Item> 880
  - Select Attribute Type 887
  - Select Property 882
  - Set Attribute 884
  - Set Feature 884
  - Set Operation 884
  - Slideshow Properties 607
  - String Viewer 1879
  - Testpoint Editor 2195
  - UML Types 981
  - Version Control Settings 346
- DIB Data Access Violation 1839
- Dictionary
  - User (Spell Checker) 486
- Diff Utility 394, 401
- Differencing
  - Facility 401
  - Output 406
  - With Baselines 394, 401
- Digraph Diagram Layout 778
- Direct Substitution
  - Field Substitution Macros, Code Template Syntax 1413
- Direct Substitution Macros 1406
- Direction Indicator
  - Connector Label 760
- Directory Structure
  - Import, Reverse Engineering 1795
- Disable
  - Add-Ins 2593
  - ArchiMate 1621
  - BPEL 1570
  - BPMN 1545
  - Data Flow Diagrams 1506
  - Entity Relationship Diagram 1627
  - Eriksson-Penker MDG Technology 1623
  - Gang Of Four Pattern Technology 1928
  - GoF Pattern Technology 1928
  - ICONIX 1925
  - MDG Technologies 1285
  - Mind Mapping 1503
  - Security 253
  - SoaML 2088
  - SOMF 2092
  - SPEM 523
  - SysML 1990
- Disable Recording Markers 1869
- Disconnect
  - Controlled Package 428
- Discussion Forum
  - Now Team Review 279
- Display
  - Attributes, Inherited 745
  - Connector Properties, Shape Scripts 1373
  - Constraints, Inherited 745
  - Element Properties, Shape Scripts 1373
  - Full Screen 129
  - Inherited Attributes 895
  - Inherited Operation 908
  - Operations, Inherited 745
  - Requirements, Inherited 745
  - Tagged Values, Inherited 745
- Display Options
  - Connector 559
  - Diagram 549
  - Element 556
  - Link 559
  - Relationship 559
- Display Size
  - Diagram, Increase 550
- Distributed Development
  - Replication 243
  - XMI Import/Export 243
- Diverge Diagram Layout 784
- Dock

- Dock
  - Two Elements 837
- Dock Windows
  - Navigation Compass 135
- Dockable
  - Element Option 837
  - Windows 132
- Docked Windows
  - For Requirements 1488
- Document
  - Enterprise Architect Content 2249
  - Exclude Packages 2249
  - HTML 2249
  - Linking 928
  - Projects 2249
  - Resource, RTF Generator (Enhanced) 2259
  - Rich Text Format 2249
  - RTF 2249
  - Single Element, RTF (Legacy) 2319
  - WSDL Element 2083
- Document Artifact
  - Element 928, 1191
  - Link Into RTF Report 928, 2309
- Document File
  - Link To From Learning Center Profile 1351
- Document Generation
  - Enterprise Architect 9
- Document Generator Interface
  - Automation Interface 2561
- Document Options
  - RTF Generator, From Diagram (Enhanced) 2255, 2310, 2312
  - RTF Generator, From Diagram (Legacy) 2255, 2310, 2312
  - RTF Report 2261
- Document Template Designer 2270
  - Commands 2275
  - Content Panel 2272
  - Context Menu 2272
  - Description 2272
  - Redo Edit 2275
  - Section Markers 2272
  - Sections Panel 2272
  - Toolbar 2272
  - Undo Edit 2275
- Documentation
  - Elements 2327
  - Generate, Project Browser Option 583
  - Group, Toolbox 2327
  - HTML 2352
  - Rich Text Format 2251
  - RTF 2251
- Documentation (Reports)
  - Submenu (Project Menu) 101
- Documentation Submenu
  - Package Context Menu, Project Browser 575
  - Reports 575
- DocumentationGeneratorInterface Package
  - DocumetationGenerator Class, Automation Interface 2562
- DocumentBreak Enum
  - Automation Interface 2416
- DocumentGenerator Class
  - Automation Interface 2562
- DocumentType Enum
  - Automation Interface 2416
- DoDAF-MODAF
  - MDG Technology For, Enterprise Architect 1289
- Domain
  - Model Template 662, 667
  - Organizational Relationships 662, 667
  - Physical Units 662, 667
  - Structure 662, 667
- DOORS, Telelogic
  - MDG Link For, Enterprise Architect 1289
- Dot
  - On Association 964
- Download
  - Gang Of Four Patterns 1273
  - GoF Patterns 1273
- Drag
  - Element From Project Browser Onto Diagram 733
  - Objects From Project Browser Onto Diagram 733
  - Text Into Diagram To Create Element 1473
- Drawing Methods
  - Shape Scripts 1366
- Drill Down
  - Composite Element 825
- Drop
  - Classifiers As Links 733
  - Classifiers As New Instances 733
  - Elements From Project Browser Onto Diagram 733
  - Objects From Project Browser Onto Diagram 733
- DTD
  - Validate XMI Import/Export 422
- Duplicate
  - Diagram 742
  - Element 808
  - Package 683

|                                               |      |  |  |
|-----------------------------------------------|------|--|--|
| Duplicate                                     |      |  |  |
| UML Diagram                                   | 742  |  |  |
| Duration                                      |      |  |  |
| Constraint                                    | 1238 |  |  |
| Constraint Between Messages                   | 1238 |  |  |
| Observation                                   | 1238 |  |  |
| Dynamic View                                  | 676  |  |  |
| Dynamic Visual Filters                        | 629  |  |  |
| <br>                                          |      |  |  |
| <b>- E -</b>                                  |      |  |  |
| EA_AddinLicenseGetDescription                 |      |  |  |
| Add-In License Management Event, Add-In Model | 2607 |  |  |
| EA_AddinLicenseValidate                       |      |  |  |
| Add-In License Management Event, Add-In Model | 2606 |  |  |
| EA_Connect                                    |      |  |  |
| Add-In Event                                  | 2597 |  |  |
| EA_Disconnect                                 |      |  |  |
| Add-In Event                                  | 2598 |  |  |
| EA_FileClose                                  |      |  |  |
| Broadcast Events, Add-In Model                | 2615 |  |  |
| EA_FileNew                                    |      |  |  |
| Broadcast Events, Add-In Model                | 2615 |  |  |
| EA_FileOpen                                   |      |  |  |
| Broadcast Events, Add-In Model                | 2616 |  |  |
| EA_GetCompartmentData                         |      |  |  |
| Compartment Events, Add-In Model              | 2609 |  |  |
| EA_GetMenuItems                               |      |  |  |
| Add-In Event                                  | 2598 |  |  |
| EA_GetMenuState                               |      |  |  |
| Add-In Event                                  | 2599 |  |  |
| EA_MenuClick                                  |      |  |  |
| Add-In Event                                  | 2601 |  |  |
| EA_OnAttributeTagEdit                         |      |  |  |
| Tagged Value Broadcast Events, Add-In Model   | 2659 |  |  |
| EA_OnConnectorTagEdit                         |      |  |  |
| Tagged Value Broadcast Events, Add-In Model   | 2660 |  |  |
| EA_OnContextItemChanged                       |      |  |  |
| Context Item Events, Add-In Model             | 2612 |  |  |
| EA_OnContextItemDoubleClicked                 |      |  |  |
| Context Item Events, Add-In Model             | 2613 |  |  |
| EA_OnDeleteTechnology                         |      |  |  |
| Technology Events, Add-In Model               | 2667 |  |  |
| EA_OnElementTagEdit                           |      |  |  |
| Tagged Value Broadcast Events, Add-In Model   | 2661 |  |  |
| EA_OnEndValidation                            |      |  |  |
| Model Validation Broadcasts, Add-In Model     | 2623 |  |  |
| EA_OnImportTechnology                         |      |  |  |
| Technology Events, Add-In Model               | 2668 |  |  |
| EA_OnInitializeTechnologies                   |      |  |  |
| Technology Events, Add-In Model               | 2663 |  |  |
| EA_OnInitializeUserRules                      |      |  |  |
| Model Validation Broadcasts, Add-In Model     | 2622 |  |  |
| EA_OnNotifyContextItemModified                |      |  |  |
| Context Item Events, Add-In Model             | 2614 |  |  |
| EA_OnOperationTagEdit                         |      |  |  |
| Tagged Value Broadcast Events, Add-In Model   | 2662 |  |  |
| EA_OnOutputItemClicked                        |      |  |  |
| Add-In Event                                  | 2602 |  |  |
| EA_OnOutputItemDoubleClicked                  |      |  |  |
| Add-In Event                                  | 2603 |  |  |
| EA_OnPostActivateTechnology                   |      |  |  |
| Technology Events, Add-In Model               | 2665 |  |  |
| EA_OnPostCloseDiagram                         |      |  |  |
| Broadcast Events, Add-In Model                | 2617 |  |  |
| EA_OnPostInitialized                          |      |  |  |
| Broadcast Events, Add-In Model                | 2617 |  |  |
| EA_OnPostNewAttribute                         |      |  |  |
| Post-New Events, Add-In Model                 | 2639 |  |  |
| EA_OnPostNewConnector                         |      |  |  |
| Post-New Events, Add-In Model                 | 2636 |  |  |
| EA_OnPostNewDiagram                           |      |  |  |
| Post-New Events, Add-In Model                 | 2637 |  |  |
| EA_OnPostNewDiagramObject                     |      |  |  |
| Post-New Events, Add-In Model                 | 2638 |  |  |
| EA_OnPostNewElement                           |      |  |  |
| Post-New Events, Add-In Model                 | 2635 |  |  |
| EA_OnPostNewGlossaryTerm                      |      |  |  |
| Post-New Events, Add-In Model                 | 2642 |  |  |
| EA_OnPostNewMethod                            |      |  |  |
| Post-New Events, Add-In Model                 | 2640 |  |  |
| EA_OnPostNewPackage                           |      |  |  |
| Post-New Events, Add-In Model                 | 2641 |  |  |
| EA_OnPostOpenDiagram                          |      |  |  |
| Broadcast Events, Add-In Model                | 2618 |  |  |
| EA_OnPostTransform                            |      |  |  |
| Broadcast Events, Add-In Model                | 2619 |  |  |
| EA_OnPreActivateTechnology                    |      |  |  |
| Technology Events, Add-In Model               | 2664 |  |  |
| EA_OnPreDeleteAttribute                       |      |  |  |
| Pre-Deletion Events, Add-In Model             | 2644 |  |  |
| EA_OnPreDeleteConnector                       |      |  |  |
| Pre-Deletion Events, Add-In Model             | 2646 |  |  |
| EA_OnPreDeleteDiagram                         |      |  |  |
| Pre-Deletion Events, Add-In Model             | 2647 |  |  |

- EA\_OnPreDeleteDiagramObject
  - Pre-Deletion Events, Add-In Model 2647
- EA\_OnPreDeleteElement
  - Pre-Deletion Events, Add-In Model 2643
- EA\_OnPreDeleteGlossaryTerm
  - Pre-Deletion Events, Add-In Model 2649
- EA\_OnPreDeleteMethod
  - Pre-Deletion Events, Add-In Model 2645
- EA\_OnPreDeletePackage
  - Pre-Deletion Events, Add-In Model 2648
- EA\_OnPreDeleteTechnology
  - Technology Events, Add-In Model 2666
- EA\_OnPreExitInstance
  - Broadcast Events, Add-In Model 2619
- EA\_OnPreNewAttribute
  - Pre-New Events, Add-In Model 2655
- EA\_OnPreNewConnector
  - Pre-New Events, Add-In Model 2652
- EA\_OnPreNewDiagram
  - Pre-New Events, Add-In Model 2653
- EA\_OnPreNewDiagramObject
  - Pre-New Events, Add-In Model 2654
- EA\_OnPreNewElement
  - Pre-New Events, Add-In Model 2651
- EA\_OnPreNewGlossaryTerm
  - Pre-New Events, Add-In Model 2658
- EA\_OnPreNewMethod
  - Pre-New Events, Add-In Model 2656
- EA\_OnPreNewPackage
  - Pre-New Events, Add-In Model 2657
- EA\_OnRetrieveModelTemplate
  - Broadcast Events, Add-In Model 2620
- EA\_OnRunAttributeRule
  - Model Validation Broadcasts, Add-In Model 2627
- EA\_OnRunConnectorRule
  - Model Validation Broadcasts, Add-In Model 2626
- EA\_OnRunDiagramRule
  - Model Validation Broadcasts, Add-In Model 2625
- EA\_OnRunElementRule
  - Model Validation Broadcasts, Add-In Model 2624
- EA\_OnRunMethodRule
  - Model Validation Broadcasts, Add-In Model 2628
- EA\_OnRunPackageRule
  - Model Validation Broadcasts, Add-In Model 2625
- EA\_OnRunParameterRule
  - Model Validation Broadcasts, Add-In Model 2629
- EA\_OnStartValidation
  - Model Validation Broadcasts, Add-In Model 2623
- EA\_QueryAvailableCompartments
  - Compartment Events, Add-In Model 2608
- EA\_ShowHelp
  - Add-In Event 2604
- EAB File
  - Export 371
  - Import 372
  - Model Branch File 371
- EABase
  - As Source 190
  - Project 47
  - Project File 189
- EAEditionTypes Enum
  - Automation Interface 2417
- EAExample File 181
- EAP File
  - As Project Database 178
  - Corrupt 540
  - Iterate Through, Automation Interface Code Example 2570
- EASL
  - Behavioral Model Templates 1449
  - Code Generation Macros, Behavioral Model 1449
  - Enterprise Architect Simulation Library 1449
- EASL Collections
  - Action 1451
  - Behavior 1451
  - Classifier 1451
  - Construct 1451
  - Node 1451
  - State 1451
  - State Machine 1451
  - Transition 1451
  - Trigger 1451
  - Vertex 1451
- EASL Properties
  - Action 1454
  - Argument 1454
  - Behavior 1454
  - Call Event 1454
  - ChangeEvent 1454
  - Classifier 1454
  - Condition 1454
  - Construct 1454
  - Edge 1454
  - EventObject 1454
  - Instance 1454
  - Parameter 1454

- EASL Properties
  - Primitive 1454
  - PropertyObject 1454
  - SignalEvent 1454
  - State 1454
  - StateMachine 1454
  - TimeEvent 1454
  - Transition 1454
  - Trigger 1454
  - Vertex 1454
- EASL\_GET
  - Code Generation Macro, Behavioral Model 1449
- EASLList
  - Code Generation Macro, Behavioral Model 1449
- ECF
  - Value 518
  - Weighting 518
- Eclipse
  - MDG Integration For, Enterprise Architect 1289
  - MDG Link For, Enterprise Architect 1289
- Edit
  - Attribute Name, In-Place Editor 918
  - Element Name, In-Place Editor 918
  - Item In Team Review 291
  - Linked Document Template 934
  - Linked Documents 930
  - Menu 93
  - Operation Name, In-Place Editor 918
  - Pattern Default 1274
  - RTF Template 2270
  - Test Details 2214
- Editions
  - Business and Software Engineering 22
  - Corporate 22
  - Desktop 22
  - Floating Licence 22
  - Lite 26
  - Model Simulation 2102
  - Of Enterprise Architect, Available 22
  - Of Enterprise Architect, Introduction 20
  - Professional 22
  - Standalone 22
  - System Engineering 22
  - Ultimate 22
- Editor
  - Team Review 291
- Editor Language Properties
  - Code Editor Properties 1896
  - Global Options 1896
  - Language-Specific Options 1896
  - Macro Key Assignment 1896
- Effort
  - Attributes 2477
  - Automation Interface, Element Package 2477
  - Field Substitution Macros 1427
  - Methods 2477
- Effort Management 452
- Effort Types 456
  - Define 456
  - Global 456
  - Non-Global 452
- EJB
  - Entity Bean Transformations 1694
  - Session Bean Transformations 1694
- Element 851, 1150, 1155
  - Abort Edit Changes 110
  - Accept Edit Changes 110
  - Action 1091
  - Activity 1101
  - Activity Final 1127
  - Activity Partition 1148
  - Activity Region 1150
  - Actor 1107
  - Add And Manage, Automation Interface Code Example 2572
  - Add Attribute, In-place Editor 924
  - Add Directly To Package 798
  - Add Link To Team Review Post 292
  - Add New Item (Inline Features Menu Option) 110
  - Add Operation, In-place Editor 924
  - Add Supporting Diagrams and Elements 832
  - Add Tagged Value (Option) 109
  - Add To Diagram 53
  - Add To Diagram From Project Browser 733
  - Add To Diagram Via Context Menu 690
  - Add To Favorites 833
  - Add To Profile 1297
  - Add To UML Model, Quick Start 53
  - Advanced Settings 847
  - Align 112, 840
  - Align From Diagram Toolbar 145
  - Align Multiple 809
  - Alternative Image 755
  - Appearance, Context Menu Option 837
  - Appearance, Format From Toolbar 692
  - Apply Image From Clipboard 112
  - ArcGIS 1634
  - ArchiMate 1621
  - Artifact 1179, 1180
  - Associated Files 875
  - Attributes Option 836
  - Author 845
  - Auto Counters, Set Up 799

- 
- Element 851, 1150, 1155
    - Auto Naming, Set Up 799
    - Auto Numbering, Set Up 799
    - Automatic Properties, C# 893
    - Automation Interface, Element Package 2478
    - Autosize Group 750
    - Autosize Single 750
    - Background Color 546, 548
    - Behavioral Diagram 1090
    - Boundary 1663
    - Boundary, Settings 1171
    - BPMN, Change Appearance 1561
    - Browser Window 876
    - Bulk Update 529
    - Business Rule 1527
    - Cardinality (Multiplicity), Non-Displayable 846
    - Central Buffer Node 1107
    - Change 2244, 2246
    - Change Appearance, BPMN 1561
    - Change Type 109, 809
    - Changes And Issues 2237
    - Changes, Add/Modify/Delete 2238
    - Child Validation 2208
    - Choice 1108
    - Class 1181
    - Code Engineering Menu 836
    - Collaboration 1184
    - Collaboration Use 1186
    - Combined Fragment 1110
    - Compartments 842
    - Complexity 845
    - Component 1187
    - Composite 825
    - Concurrency 846
    - Confirm As Parent 805
    - Connect 943
    - Connectors In Relationships Window 648
    - Constraint Note 1147
    - Constraint, Attach 110
    - Constraints 850
    - Context Menu 827
    - Context Menu, Add Submenu 832
    - Context Menu, Advanced 829
    - Context Menu, Features 836
    - Context Menu, Find Submenu 833
    - Context Menu, Project Browser 577
    - Continuation 1155
    - Continutaion 1155
    - Control 1665
    - Coordinates On Status Bar 152
    - Copy And Paste Between Diagrams 57
    - Copy Between Packages 808
    - Create Child Diagram 724
    - Create From Linked Document 932
    - Create From Maintenance Item 2241
    - Create From Text File 797
    - Create From Text File Text 1473
    - Create From Toolbox 698
    - Create In Diagram 797
    - Create Link From Project Browser 952
    - Create With Quick Linker 792
    - Create, Same Type As Previous 105
    - Cross References (Option) 109
    - Customize Visibility 813
    - Cut And Paste Between Diagrams 57
    - Dashed Border On 805
    - Data Store, Data Flow Diagram 1506
    - Data Type 1188
    - Datastore, Activity Diagram 1116
    - Decision 1117
    - Default Appearance 112
    - Default Element Template 820
    - Defect 2244
    - Defects 2237
    - Defects, Add/Modify/Delete 2238
    - Delete 811
    - Delete Item From 110
    - Deployment Spec 1189
    - Details 846
    - Device 1190
    - Diagram Frame 1119
    - Diagram Gate 1120
    - Display Depth 744
    - Display Options 556
    - Dockable 837
    - Document Artifact 1191
    - Documentation 2327
    - Drag From Project Browser 733
    - Duplicate 808
    - Edit Attribute Keyword 921
    - Edit Attribute Scope 920
    - Edit Attribute Stereotype, In-place Editor 919
    - Edit Item, Tasks 916
    - Edit Name, In-Place Editor 918
    - Edit Operation Parameter Keyword 922
    - Edit Operation Parameter Kind 923
    - Edit Operation Scope 920
    - Edit Operation Stereotype, In-place Editor 919
    - Embedded 109
    - Embedded Submenu 834
    - Embedded, Add 835
    - Endpoint 1121
    - Entity 1666
    - Entity, Entity Relationship Diagram 1627

- Element 851, 1150, 1155
  - Entry Point 1123
  - Enumeration 1191
  - Eriksson-Penker 1623
  - Event 1667
  - Exception 1123
  - Execution Environment 1192
  - Exit Point 1126
  - Expansion Region 1124
  - Export Data In CSV Format 439
  - Expose Interface 1193
  - Extended By Stereotype 1663
  - External Requirements 850
  - External, Data Flow Diagram 1506
  - Feature 1485
  - Feature, Connect To 944
  - Feature, Disconnect From 944
  - Fill Color 546, 548
  - Fill Color Gradient 550
  - Filter Display On Diagram 629, 631
  - Find In Diagram Menu Options 577
  - Find in Diagrams 108, 833
  - Find In Diagrams, Package Browser 592
  - Find In Project Browser 108, 833
  - Find Related 738
  - Fine Movement 763
  - Flow Final 1128
  - Font, Set 112
  - Fork 1130, 1131
  - Fragment 1110
  - Gang Of Four Pattern 1928
  - Gap, Model 651
  - General Settings 845
  - Get Project Custom Colors 819
  - GoF Pattern 1928
  - Hide Type On Diagram 629, 631
  - Highlight Context 822
  - History 1134
  - Hyperlink 1668, 1670
  - Icon, Project Browser, User-Defined 1308
  - Icons On Diagram 841
  - Image 1672
  - Import Data In CSV Format 440
  - Include Linked In RTF Report 2255, 2310, 2312
  - Information Item 1194
  - Initial 1136
  - In-place Editor Options 916
  - In-Place Formatting 841
  - Insert Maintenance Feature 925
  - Insert New Feature (Inline Features Menu Option) 110
  - Insert Operation Parameter 924
  - Insert Related Elements 833
  - Insert Testing Features 926
  - Instance 1196
  - Interaction 1137
  - Interaction Occurrence 1139
  - InteractionUse 1139
  - Interface 1194
  - Internal Requirements 848
  - Interruptible Activity Region 1141
  - Issue 2244
  - Issues And Changes 2237
  - Issues, Add/Modify/Delete 2238
  - Join 1130, 1133
  - Junction 1142
  - Keywords 845
  - Labels 760
  - Layout 840
  - Legend 748
  - Lifeline 1144
  - Line Color 548
  - Link To Attribute Via Object 896
  - Linked, Convert To Local Copy 823
  - Lock Indicators 276
  - Lock, Require User Lock 274
  - Lock, Security Off 837
  - Lock, User/Group Lock 271
  - Locked, Add Connector To 273
  - Maintenance 2237
  - Make All Selectable/Unselectable 687
  - Make Non-Selectable, Multiple Elements 840
  - Make Non-Selectable, Single Element 837
  - Make Same Size 112
  - Make Selectable, Multiple Elements 840
  - Make Selectable, Single Element 837
  - Managing 795
  - Master Document 2327
  - Match Size 840
  - Menu 108
  - Merge 1117
  - Merge Node 1145
  - Mind Mapping 1503
  - Model Document 2327
  - Modeling With 795
  - Move Between Packages 806
  - Move By Increments 763
  - Move In Diagrams 805
  - Move, Impact Of Diagram 57
  - Multiple Selection 840
  - Multiple Update 529
  - N-Ary Association 1673
  - N-Ary, Entity Relationship Diagram 1627
  - New 797

- Element 851, 1150, 1155
  - Node 1196
  - Non-Selectable, Multiple Elements 840
  - Non-Selectable, Single Element 837
  - Note (Constraint, Comment) 1147
  - Note, Attach 110
  - Notes 814, 977
  - Nudge 112, 763
  - Object 1196
  - Occurrence 1139
  - Operations Option 836
  - Overrides & Implementations 109
  - Package 1200
  - Packaging Component 1674
  - Parents And Interfaces 109
  - Part 1200
  - Partition 1148
  - Paste As Link 94
  - Paste As New 94
  - Paste Copy 808
  - Paste From Clipboard As Metafile 94
  - Paste From Project Browser 733
  - Phase 845
  - Phase, Update For Package 529
  - Place Related On Current Diagram 738
  - Port 1202
  - Position Options 108
  - Postconditions 850
  - Preconditions 850
  - Primitive 1204
  - Process 1518, 1674
  - Process, Data Flow Diagram 1506
  - Properties Dialog 843
  - Properties Dialog, Changes and Defects 2247
  - Properties Dialog, General Settings 845
  - Properties Menu 828
  - Properties Window 878
  - Properties, As Attributes 892
  - Properties, Edit From Package 1011
  - Properties, Links 851
  - Pseudo-State 1037
  - Receive 1149
  - Receive Event 1667
  - Region 1150
  - Region, Expansion 1124
  - Region, Interruptible Activity 1141
  - Relationship Traceability For 637
  - Relationship, Delete 851
  - Relationship, Entity Relationship Diagram 1627
  - Relationship, Hide 851
  - Relationship, Show 851
  - Requirement 1471, 1483
  - Requirements 848
  - Resize 810
  - Responsibilities 848
  - Risk 1675
  - Rule Task 1530
  - Scenario 852
  - Scenarios & Requirements 879
  - Screen 1656
  - Select By Property 629
  - Selectable, Multiple Elements 840
  - Selectable, Single Element 837
  - Send 1151
  - Send Event 1667
  - Sequence Diagram 1078
  - Sequence, Lifecycle 1076
  - Set Alternative Image 112
  - Set Cross References 803
  - Set Custom References 803
  - Set Default Appearance 817
  - Set Element Template Package 820
  - Set Feature Visibility 836
  - Set Font 839
  - Set Parent 801
  - Set Project Custom Colors 819
  - Shadow Color 546
  - Shape Script Properties 1373
  - Show Usage 802
  - Signal 1205
  - Size 810
  - SoaML 2088
  - SOMF 2092
  - SPEM 524
  - State 1152
  - State Invariant 1155, 1157
  - State Lifeline 1158
  - State Machine 1161
  - State/Continuation 1155
  - Status 845
  - Status, Update For Package 529
  - Stereotype 845
  - Stored Procedure 1961
  - Structural Diagram 1179
  - Structured Activity, Conditional Node 1161, 1168
  - Structured Activity, Loop Node 1161, 1164
  - Structured Activity, Sequential Node 1161, 1164
  - Structured Activity, Structured Node 1161, 1163
  - Sub-Activity 1101
  - Sub-Activity, Conditional Node 1161, 1168
  - Sub-Activity, Loop Node 1161, 1164



- Element 851, 1150, 1155
  - Sub-Activity, Sequential Node 1161, 1164
  - Sub-Activity, Structured Node 1161, 1163
  - Submachine State 1152, 1161
  - Superimposition 744
  - Synch 1169
  - Synchronize Stereotyped Tagged Values From Toolbox 698
  - Synchronize Stereotypes With Profile 1281
  - SysML Activity 1992
  - SysML Block Definition 1996
  - SysML Interaction 1999
  - SysML Internal Block 2000
  - SysML Model 2002
  - SysML Parametric 2005
  - SysML Requirement 2007
  - SysML State Machine 2009
  - SysML Use Case 2011
  - System Boundary 1170
  - Table 1630
  - Task 1676
  - Tasks 795, 2237
  - Tasks, Add/Modify/Delete 2238
  - Template Package 820
  - Template Parameters 846
  - Template, Default Element 820
  - Templates And Profiles 820, 1277
  - Terminate 1172
  - Test Case 1677
  - Test Cut 2198
  - Test Scripts Compartment 2229
  - Test Set 2198
  - Test Suite 2199
  - Text 814
  - Text Color 546, 548
  - Toolbar 143
  - Transformation 1683
  - Trigger 1173
  - Type, Change 109
  - UI Control 1657
  - UML 1090
  - Usage 802
  - Use Case 1174
  - Use Circle Notation 1177
  - Use Cross References 803
  - Use Custom References 803
  - Use Extras, Automation Interface Code Example 2575
  - Use Rectangle Notation 1177
  - User Interface 1657
  - Value Lifeline 1177
  - Version 845
  - Version, Update For Package 529
  - View Properties (Inline Features Menu Option) 110
  - Visibility 846
  - Visibility Options 559
  - Visual Representation, Introduction 841
  - Work On From Diagram List 596
  - Work On From Package Browser 587
  - Work On From Toolbar 146
  - Working With 795
  - WSDL Binding 2078
  - WSDL Message 2073
  - WSDL Namespace 2067, 2070
  - WSDL Service 2082
  - WSDL, Document 2083
  - WSDL, Port Type 2075
  - Z-Order 112, 744
- Element Context Menu
  - Add Submenu, Project Browser 580
- Element Feature
  - Attribute 885
  - Connect To 944
  - Disconnect From 944
  - Operation 900
- Element Package, Automation Interface
  - Constraint 2476
  - Diagram 2474
  - Effort 2477
  - Element 2478
  - File 2491
  - Issue 2492
  - Metric 2494
  - Requirement 2495
  - Resource 2497
  - Risk 2498
  - Scenario 2499
  - ScenarioExtension 2501
  - ScenarioStep 2502
  - TaggedValue 2504
  - Test 2506
- Element Templates
  - And Profiles 1293
- Element View
  - Project Task Allocation 475
- ElementFeatures Package, Automation Interface 2410
  - Attribute Class 2392
  - AttributeConstraint 2396
  - AttributeTag 2397
  - CustomProperties Collection 2399
  - Diagram 2391
  - EmbeddedElements Collection 2399

- ElementFeatures Package, Automation Interface 2410
  - Method 2400
  - MethodConstraint 2404
  - MethodTag 2405
  - Parameter 2406
  - ParamTag 2408
  - Partitions Collection 2409
  - Properties 2410
  - Property 2410
  - Transitions Collection 2412
- Ellipse Diagram Layout 770
- Email
  - Access Within Enterprise Architect 165
  - Exchange Server, Define Default 544
- Embedded Elements
  - Dialog 835
  - Element Context Submenu 834
  - In Child Composite Diagrams 826
  - Include In RTF Reports 2308
  - Include When Pasting Composite 736
  - Incorporate Inherited Properties 835
  - Maintain Layout In Pasting Composite 736
  - Submenu Option 109
  - Window 835
- EmbeddedElements Collection
  - Automation Interface, ElementFeatures Package 2399
- EMX
  - File Cross References 419
  - File Import 419
- Enable
  - Add-Ins 2593
  - ArchiMate 1621
  - Data Flow Diagrams 1506
  - Entity Relationship Diagram 1627
  - Eriksson-Penker MDG Technology 1623
  - Exclusive Diagram Edit Lock 253
  - Gang Of Four Pattern Technology 1928
  - GoF Pattern Technology 1928
  - ICONIX 1925
  - MDG Technologies 1285
  - Mind Mapping 1503
  - Security 253
  - SoaML 2088
  - SOMF 2092
  - SPEM 523
- Encrypt Password
  - Prior To Release 7.1 Of Enterprise Architect 269
- Encrypt Password (Repository)
  - At Release 7.1 Of Enterprise Architect 183, 188
- End Event, BPEL 1.1
  - Create 1586
  - Model 1586
  - Types 1586
- End Event, BPEL 2.0
  - Create 1608
  - Model 1608
  - Types 1608
- End User License Agreement 28
- Endpoint
  - Element 1121
- Enterprise Architect
  - Add-In Model 2585
  - Alignment With UML 1008
  - And Deployment 81
  - And IDEs 9
  - Autohide Windows 137
  - Automation 2371
  - Build Systems 9
  - CASE Tool 3
  - Change Control, Project 2233
  - Change Management 311
  - Code Engineering With 1736
  - Community Site 40, 87
  - Connectors 1206
  - Create Project, Tutorial 49
  - Database Modeling 9
  - Debug 9
  - Demonstration 86
  - Design Systems 9
  - Dock Windows 135
  - Dockable Windows 132
  - Editions, Differences Between 22
  - Editions, Introduction 20
  - Editor 1895
  - End User License Agreement 28
  - Example Project File 181
  - Export Data In CSV Format 439
  - Extend Trial Period 20
  - Feedback Pages 3
  - For Business Analysts 74
  - For Database Administrators 84
  - For Developers 78
  - For Implementation Managers 81
  - For Project Managers 80
  - For Software Architects 75
  - For Software Engineers 77
  - For Technology Developers 82
  - For Testers 80
  - Formal Statements 27

- Enterprise Architect
  - Fundamental Processes 7
  - Generate Documentation 9
  - Generate Source Code 9
  - Getting Started 44
  - Glossary 2689
  - Help 40
  - How To Use 3
  - Import Data In CSV Format 440
  - In Action 86
  - Install 36
  - Interfaces For Modeling 86
  - Introduction 3
  - Key Features 14
  - Keyboard Shortcuts 166
  - Keyboard/Mouse Shortcuts 174
  - License Agreement 28
  - Lite Edition 26
  - Main Context Menu 90
  - Main Menu 90
  - Manage Model Structure 9
  - Manage Requirements 9
  - MDA Transformation 9
  - Model Complexity 9
  - Model Requirements 9
  - Modeling Tool 3
  - Navigation Compass 135
  - Object Model, Introduction 2381
  - Online Resources 87
  - Online User Guide 3
  - Order 35
  - Performance, DBMS Connections 210, 212, 214, 216, 218, 219, 221
  - Performance, In Auditing 382
  - Performance, WAN Optimizer 238
  - Predefined Search Definitions 617
  - Pricing And Purchasing 35
  - Professional Roles 71
  - Project Change Control 2233
  - Project Files, Open A Project 181
  - Project Maintenance 2233
  - Project Roles 71
  - Project Team Review 279
  - Project, What Is A? 179
  - Quality Control 2202
  - Quickstart Tutorial 49
  - Read-Only Edition 26
  - Register Full License 38
  - Replication Merge Rules 244
  - Reverse Engineer 9
  - Scripting 2371
  - SDK, Introduction 1292
  - Select Edition To Trial 20
  - Share Model Development 9
  - Software Product License Agreement 28
  - Spell Checking 480
  - Start 47
  - Start Page 87
  - Support 40, 42
  - Testing 2202
  - Tools And Features For Modeling 86
  - Trial Version 20
  - UML 2.3 Support 9
  - UML Modeling With 657
  - User Interface 86
  - Uses Of 9
  - Visualize Systems 9
  - Web Services Access 165
  - What Can I Do With It? 9
  - What is Enterprise Architect? 5
  - Working With 7
  - Workspace Status Bar 152
- Enterprise Architect Lite
  - Availability 26
  - Description 26
  - Edition 26
- Enterprise Architect Simulation Library
  - Behavioral Model Templates 1449
  - EASL Code Generation 1449
  - EASL\_GET Macro 1449
  - EASLList Macro 1449
- Enterprise Architect Toolbox
  - Documentation Group 2327
  - MDG Technology Groups 1284
  - SystemC Group 1760
  - Verilog Group 1763
  - VHDL Group 1764
- Enterprise Java Beans 1694
  - MDG Technology For, Enterprise Architect 1289
- Entity
  - Create 1667
  - Element 1666
  - Element (Entity Relationship Diagram) 1627
- Entity Relationship Diagram
  - Concepts 1627
  - Connector 1627
  - Elements 1627
  - Example Diagram 1627
  - MDG Technology 1627
  - Relationship 1627
  - Tagged Values 1627
  - Toolbox Page 1627
  - Transformation From Data Model 1689

- Entity Relationship Diagram
  - Transformation To Data Model 1697
- Entry Point
  - Element 1123
- Entry Points
  - Tab Of Structural Specification 854, 863
- Enumeration
  - Automation Interface 2413
  - ConstLayoutStyles 2414
  - CreateBaselineFlag 2416
  - CreateModelType 2416
  - DocumentBreak 2416
  - DocumentType 2416
  - EAEditionTypes 2417
  - Element 1191
  - EnumRelationSetType 2417
  - ExportPackageXMIFlag 2418
  - Literal 1191
  - MDGMenus 2418
  - MessageFlag 2419
  - ObjectType 2419
  - PropType 2421
  - ReloadType 2421
  - ScenarioDiagramType 2422
  - ScenarioStepType 2422
  - ScenarioTestType 2423
  - XMIType 2423
- Enumeration Elements
  - Add To Profiles 1303
- EnumRelationSetType Enum
  - Automation Interface 2417
- Environment Complexity Factor
  - Definition 518
  - Estimate Project Size 520
  - Estimation 518
  - Value 518
  - Weighting 518
- ERD To Data Model
  - Transformation, MDA-Style Transform 1697
- ERDs 1627
- Eriksson-Penker
  - Concept 1623
  - Diagram 1623
  - Disable 1623
  - Elements 1623
  - Enable 1623
  - Extensions 1623
  - MDG Technology 1623
  - Relationships 1623
  - Toolbox Page 1623
- Eriksson-Penker Business Extensions 1511
- Establish Port-Trigger Mappng
  - Model State Machine For HDL 1784
- Estimation 1000
  - Default Hours 520
  - Environment Complexity Factors 518
  - Of Project Factors 516
  - Of Project Size 520
  - Of Project Timescale 516
  - Technical Complexity Factors 517
  - Use Case 516
- Estimation Factors
  - Project Types Menu Option 129
- Event
  - Business Modeling 1520
  - Created In Project Calendar 505
  - Displayed In Project Calendar 505
  - Element 1667
  - Receive 1667
  - Recurring, In Project calendar 505
  - Send 1667
- EventProperties
  - Automation Interface Repository 2431
- EventProperty
  - Automation Interface Repository 2432
- Example Diagram
  - Entity Relationship Diagram 1627
- Examples And Tips
  - Automation Interface 2386
- Exception
  - Element 1123
- Exception Path
  - Add To Scenario (Toolbar) 857
- Exception Path, Scenario 852
- Exclamation Mark
  - Blue 276
  - Red 276
- Exclude
  - Package In Generated Report 2260
- Exclusive Edit Lock
  - Automatic 253
  - Disable 253
  - Enable 253
  - Toggle 253
- Exclusive Gateway, BPEL 1.1
  - Data-Based 1584
  - Event-Based 1584
- Exclusive Gateway, BPEL 2.0
  - Data-Based 1607
  - Event-Based 1607
- Execution Analysis
  - Breakpoints And Markers Window, Record Sequence Diagrams 1869, 2157

- Execution Analysis
  - Control Recording, Record Sequence Diagrams 2161
  - Control Stack Depth 2153
  - Diagram Features, Generate Sequence Diagrams 2152
  - Difference Between Recording Marker And Breakpoint 2155
  - Generate Sequence Diagram 2164
  - Marker Types, Record Sequence Diagrams 2155
  - Object Workbench, Create Variables 2177
  - Object Workbench, Introduction 2176
  - Object Workbench, Invoke Methods 2179
  - Object Workbench, Overview 2176
  - Place Markers, Recording Sequence Diagrams 2154
  - Profiler Operation 2174
  - Profiler Report, Load 2175
  - Profiler Report, Save 2175
  - Profiler Report, Save As Resource In Team Review 2175
  - Profiler Toolbar 2171
  - Profiler, Attach To Process 2173
  - Profiler, Getting Started 2171
  - Profiler, Launch 2173
  - Profiler, Overview 2170
  - Profiler, Prerequisites 2171
  - Profiler, Set Options 2174
  - Profiler, Set Sample Intervals 2174
  - Profiler, Start 2173
  - Profiler, Stop 2173
  - Profiler, Supported Platforms 2171
  - Profiler, System Requirements 2171
  - Record Activity For Method 2155
  - Record Sequence Diagrams, Introduction 2147
  - Record Sequence Diagrams, Overview 2148
  - Record Sequence Diagrams, Set Up 2153
  - Record State Transitions 2165
  - Record Unit Test Results 2185
  - Recording Activity For Class 2160
  - Recording History 2149
  - Recording Markers, Activate, Record Sequence Diagrams 1869
  - Recording Markers, Disable, Record Sequence Diagrams 1869
  - Run Unit Test 2185
  - Save Recording History 2164
  - Sequence Diagrams, Limit Auto Recording 2155
  - Set Up To Capture State Changes 2166
  - State Machine 2166
  - State Transitions 2166
  - Team Review, Save Profiler Report As Resource 2175
  - Unit Test Script, Set Up 2183
  - Unit Test, Record Results 2185
  - Unit Testing, Introduction 2183
  - Work With Marker Sets, Record Sequence Diagrams 2158
- Execution Analyzer
  - Context Menu 1828
  - Options 1828
  - Record Sequence Diagrams, Recording Options 1863
  - Recording 2161
  - Toolbar 1828
  - Window 1828
- Execution Environment
  - Element 1192
- Execution Profiler
  - Attach To Process 2173
  - Getting Started 2171
  - Launch 2173
  - Operation 2174
  - Overview 2170
  - Prerequisites 2171
  - Report, Example 2170
  - Report, Load 2175
  - Report, Save 2175
  - Report, Save As Resource In Team Review 2175
  - Set Options 2174
  - Set Sample Intervals 2174
  - Start 2173
  - Stop 2173
  - Supported Platforms 2171
  - System Requirements 2171
  - Team Review, Save Report As Resource 2175
  - Toolbar 2171
- Exit
  - Menu Option (File Menu) 92
- Exit Point
  - Element 1126
- Expansion Node
  - Action 1099
- Expansion Region
  - Element 1124
- Export
  - .EAB File 371
  - ArcGIS Schema 1640
  - ArcGIS XML Workspace 1640
  - Code Generation Templates 1400
  - Code Templates 1462
  - Data 306

## Export

- MDA-Style Transformations 1400
- Model Elements To External Tools 414
- MOF To XMI 2097
- OWL/RDF .XML Files 1652
- Profile, To Disk 1306
- Reference Data 306
- Reference Data, Introduction 305
- RTF Template 2270
- State Machine Table To CSV 1048
- To Rational Rose 411, 413, 421
- To XMI 413, 414
- To XML 414
- Version Controlled Model Branch 371
- XMI, Batch 430

## Export Diagrams

- To RTF Document 2340

## ExportPackageXMIFlag Enum

- Automation Interface 2418

## Expose Interface

- Element 1193

## Extend

- Connector 1223
- Relationship 1223
- Toolbox Connectors 1345
- Toolbox Elements 1344

## Extended Elements 1663

## Extended UML Diagrams 1502

## Extending Modeling Languages

- And MDG Technologies 1292
- With Enterprise Architect 1292
- With UML Patterns In Enterprise Architect 1292
- With UML Profiles In Enterprise Architect 1292
- With UML Stereotypes In Enterprise Architect 1292

## Extension

- SysML Requirement 2007

## Extension Points

- Use Case 1176

## Extension Stereotypes 1663

## External

- Element (Data Flow Diagram) 1506
- Requirements 850, 1483

## External File

- Add Link To Team Review Post 292

## External Requirements

- Color Coded 1477

## External Tools

- Open 157
- Pass Parameters To 159

**- F -**

## Fan Relations Diagram Layout 783

## Favorites

- Add To Resources Window 1005
- Delete From Resources Window 1005
- Drag Objects Into 598, 604
- Elements, Resources Window 1005
- Folder, Resources Window 1005
- Model Views Folder 598
- View Properties In Resources Window 1005

## FDD Methodology 1485

## Feature

- As Attribute 885
- As Operation 900
- Element 1485

## Feature Driven Design Methodology 1485

## Feature Visibility

- Attributes 745
- Customize 745
- Inherited 745
- Operations 745
- Set 745
- Suppress 745

## Features

- Connect To 944
- Disconnect From 944
- Of Enterprise Architect 14

## Feedback Pages 3

## Field Substitution Macros

- Access Data From Attributes 1414
- Access Data From Classes 1416
- Access Data From Connectors 1423
- Access Data From Constraints 1426
- Access Data From Efforts 1427
- Access Data From Files 1427
- Access Data From Links 1429
- Access Data From Metrics 1431
- Access Data From Operations 1431
- Access Data From Packages 1432
- Access Data From Parameters 1433
- Access Data From Problems 1434
- Access Data From Requirements 1435
- Access Data From Resources 1436
- Access Data From Risks 1436
- Access Data From Scenarios 1437
- Access Data From Tests 1438
- Attribute 1414
- Class 1416
- Conditional Substitution 1413

- Field Substitution Macros
  - Connector 1423
  - Constraint 1426
  - Direct Substitution 1413
  - Effort 1427
  - Examples 1413
  - File 1427
  - File Import 1428
  - Import, File 1428
  - Link 1429
  - Linked File 1430
  - Metric 1431
  - Miscellaneous 1411
  - Operation 1431
  - Package 1432
  - Parameter 1433
  - Problem 1434
  - Project 1411
  - Requirement 1435
  - Resource 1436
  - Risk 1436
  - Scenario 1437
  - Source Code Generation Options 1418
  - Test 1438
- Fields and Conditions
  - In Search 629
- File
  - .EAB 371
  - Element Package, Automation Interface 2491
  - Field Substitution Macros 1427
  - Hyperlink To 1669
  - Linked, Field Substitution Macros 1430
  - Menu 92
- File Import
  - Field Substitution Macros 1428
- File Search
  - List View 1817
  - Search Window, Debugging 1817
  - Toolbar 1817
  - Tree View 1817
  - Use 1817
- Filter
  - Diagram, Display Of Connectors 631
  - Diagram, Display Of Elements 631
  - Hidden Or Visible Connectors 631
- Filter Bar
  - In Diagram List 590
  - In Model Search 619
  - In Package Browser 590
- Filters
  - Add To Search 627
  - Analyzer Script, Recording Options 1863
  - AND 621
  - Diagram, Display Of Connectors 629
  - Diagram, Display Of Elements 629
  - Dynamic Visual 629
  - Element, RTF Report 2264
  - OR 621
  - Other, RTF Report 2267
  - Search 621
- Find
  - Diagram In Project Browser 105
  - Element In All Diagrams 833
  - Element In Diagrams 108
  - Element In Diagrams, Package Browser 592
  - Element In Project Browser 108, 833
  - Submenu 833
- Find And Replace
  - Access Facility 1814
  - Dialog 1814
  - In Code Editor 1814
  - Metasequences 1814
  - Regular Expressions 1814
  - Tagged Regions 1814
- Find In Diagram
  - Element Context Menu, Project Browser 577
- Find in Project
  - Menu Option (Edit Menu) 93
- Find In Project Browser
  - Search Option 571
- First Chance Exception
  - Process 1889
- Floating Licence
  - Version of Corporate Edition 22
- Floating Toolbar Buttons
  - Display 556
- Floating Windows 135
- Flow Final
  - Element 1128
- Font
  - Set For Element Text 839
- Fonts
  - Set Model Default 552
  - Set User Default 552
- Footer
  - Insert In RTF Template 2290
- Footers
  - Add To RTF Document 2349
- Foreign Key
  - Composite, Create 1959
  - Composite, Representation In Diagram 1959
  - Constraint 1956
  - Create 1956
  - Description 1956

- Foreign Key
    - Name Template, Define 1960
    - Representation In Diagram 1956
  - Foreign Language Translation
    - RTF Report Generation 2269
  - Fork
    - Element 1130, 1131
    - Node In Simulation 2137
    - Pseudo-State 1130, 1131
  - Fork/Join
    - Element 1130
  - Formal Statements 27
  - Format
    - Element Appearance From Toolbar 692
    - Toolbar 692
  - Forum
    - Discussion, Now Team Review 279
  - Forward Engineering
    - Initial Code In Operations 906
    - Introduction 1733
  - Forward Slash
    - Derived Symbol 887
  - Forward Synchronization
    - Delete Code From Features In Model Of Package 1899 1774
  - Fragment
    - Element 1110
  - Frame
    - Combine Windows In 135
    - Remove Windows From 135
    - Tabbed 135
  - Free Sorting
    - On Project Browser 587
  - Full Screen
    - Close 129
    - Display 129
  - Fully Qualified Tagged Value
    - Show In Element Compartment 745
  - Function
    - Macros, Code Template Syntax 1439
    - Step Out Of 1875
  - Function Call
    - Step Into 1875
    - Step Through 2162
- G -**
- Gang Of Four Pattern
    - Concepts 1928
    - Download 1271, 1273
    - MDG Technology 1928
  - Gap Analysis Matrix
    - Edit Notes 654
    - Gap Elements 651
    - Notes, Edit 654
    - Overview 651
    - Profiles 651
  - Gap Element
    - Create 651
  - Gap Notes
    - Edit 654
  - Garden Of Eden Style 2051
  - Gateway, BPEL 1.1
    - Create 1584
    - Exclusive (XOR) 1584
    - Inclusive (OR) 1584
    - Model In BPEL 1.1 1584
    - Parallel (AND) 1584
    - Types 1584
  - Gateway, BPEL 2.0
    - Create 1607
    - Exclusive (XOR) 1607
    - Inclusive (OR) 1607
    - Model In BPEL 2.0 1607
    - Parallel (AND) 1607
    - Types 1607
  - GDB
    - Local 1859
    - Remote 1859
    - Script Set Up 1859
  - General
    - Types, Dialog 995
  - General Options
    - Set For Project 544
  - General Ordering
    - Sequence Diagram Messages 1240
  - General Types
    - Project Types Menu Option 129
  - Generalization
    - Sets 955
  - Generalization Link
    - Implement Parent Operations 908
    - Override Parent Operations 908
  - Generalize
    - Connector 1224
    - Relationship 1224
  - Generate
    - BPEL 1.1 1594
    - BPEL 2.0 Code 1618
    - DDL 1978
    - Global Element For Global ComplexTypes 2049
    - Global Element In XSD 2051
  - Toolbox Page 1928



- Generate
  - HTML Report From Virtual Document 2368
  - Package Source Code, (Tools Menu Option) 116
  - Report On Elements 587, 596
  - Report On Project 587
  - Rich Text Format Report (Legacy) 2323
  - RTF Document from Resources (Enhanced) 2259
  - RTF Document From Resources (Legacy) 2325
  - RTF Report (Legacy) 2323
  - RTF Report From Diagram List 596
  - RTF Report From Package Browser 587
  - RTF Report From Virtual Document 2338
  - WSDL 2085
  - XML Schema For Referenced Packages 2049
  - XML Schema, For Child Packages 2049
  - XSD 2049
- Generate Code
  - For Business Rule 1543
  - From Activity Diagrams 1778, 1787
  - From Behavioral Models 1778
  - From Interaction Diagrams 1778, 1786
  - From Sequence Diagrams 1778, 1786
  - From State Machine Diagrams 1778, 1780
  - Hardware Description Languages Supported 1778
  - Supported Hardware Description Languages 1778
- Generate HTML Report Dialog 2354
- Generate RTF Document
  - Element Filters Tab 2264
  - Other Filters Tab 2267
  - Project Constants, User Defined 2268
  - Update Links In Word 2351
- Generate RTF Documentation Dialog
  - General Tab 2256
  - Optimize For Sun Open Office 2256
  - Options 2256
  - Output To PDF 2256
  - Template Editor 2275
  - Templates Tab 2270
  - Word Substitution Tab 2269
- Generate Sequence Diagram
  - Diagram Features 2152
  - Execution Analysis 2164
- Generate Source Code
  - Enterprise Architect 9
  - Overview 1768
- Geodatabase
  - Export ArcGIS XML Workspace 1640
  - Import ArcGIS XML Workspace 1641
- Modeling 1633
- Get All Latest 357
- Get Function
  - Create As Attribute Property 892
- Get Latest 357
- Get Project Custom Colors
  - For Element 819
- Getting Started
  - Add License Key 2739
  - In Enterprise Architect 44
  - Installing Enterprise Architect 36
  - License Information 2738
  - License Management 2736
  - Register Enterprise Architect 38
  - Registration Key 2738
  - Start Enterprise Architect 47
  - Upgrade Existing License 2743
  - With MOF 2095
- Global Element
  - Generate For Global ComplexTypes 2049
  - Generate In XSD 2051
  - Import XSD 2054
- Global Elements
  - Import 2053
- Global Risks 458
- Glossary
  - A 2690
  - Add Item, Glossary Dialog 469
  - B 2693
  - C 2695
  - Create Item From Notes Text 977
  - D 2699
  - Delete Item, Glossary Dialog 469
  - Dialog 469
  - E 2702
  - F 2704
  - Filter List, Glossary Dialog 469
  - G 2706
  - H 2707
  - Hyperlink Term From Notes 977
  - I 2708
  - Insert Item In Text 977
  - J 2710
  - L 2711
  - M 2712
  - Model 467
  - Modify Item, Glossary Dialog 469
  - N 2715
  - O 2716
  - Of Terms 2689
  - P 2717
  - Project 467

- Glossary
    - Q 2721
    - R 2722
    - Report 471
    - Report Output Sample 472
    - S 2725
    - T 2730
    - U 2732
    - V 2734
  - Glossary Detail Dialog
    - Add Item 468
    - Modify Item 468
  - GNU Debugger
    - Local 1859
    - Remote 1859
    - Script Set Up 1859
  - Goal
    - Business Modeling 1521
  - GoF Pattern
    - Concepts 1928
    - Download 1271, 1273
    - MDG Technology 1928
    - Toolbox Page 1928
  - Green Circle
    - Breakpoint Marker On Element During Simulation 2114
  - Grid
    - Diagram, Adjust 553
  - Group Lock
    - Identify Owner 277
  - Group Login 262
  - Group Properties Dialog
    - Script 2375
  - Guillemets 1267
- H -**
- Hand Drawn
    - Diagram Mode 727
    - Diagram Mode, API 1348
  - Hardware Description Languages
    - Model State Machine For 1784
    - Supported For Code Generation 1778, 1784
  - HDL
    - Model State Machine For 1784
  - Header
    - Insert In RTF Template 2290
  - Headers
    - Add To RTF Document 2349
  - Help
    - Add-In 125
    - Display On Add-Ins 125
  - File Formats 41
  - For Tagged Values 973
  - Index Tab 40
  - Learning Center Topics 89
  - Menu 131
  - Release Date 41
  - Search Tab 40
  - Systems 40
  - Topic, Hyperlink To 1668
  - Version Details 41
  - Hidden
    - Connectors, Filter For On Diagram 631
  - Hidden Submenu
    - Create In Toolbox Profile 1341
  - Hide
    - Connectors 954
    - Connectors, All Diagrams 955
    - Connectors, Requirements Element 955
    - Connectors, Single Diagram 955
    - Diagram Caption Bar 98
    - Labels 957
    - Package Contents On Diagram 685
    - Project Browser 567
    - Relationship 851
    - Toolbox 698
  - Hide/Show
    - Connector Labels 941
    - Connectors 941
  - Hierarchy
    - Of Requirements 1483
    - Requirement, Import Via CSV 1481
  - Highlight
    - Context Element 822
  - History
    - Deep 1134
    - Element 1134
    - Shallow 1134
  - Hotkeys
    - Diagram Navigation And Selection 740
  - Hourly Rate 520
  - HTML
    - And CSS Style Editor 2355
    - Documentation 2249, 2352
    - Generate HTML Report Dialog 2354
    - Locate Page By GUID 2353
    - Quick Start, Generate Report 2353
    - Report 2249, 2352
    - Report Templates 2355
    - Report, Create 2353
    - Style Template Fragments 2357
    - Style Templates, Reporting 2355
    - Template Fragments 2357

- HTML
  - View Report 2353
- HTML Report
  - Exclude Package 2260
  - Generate From Virtual Document 2368
  - Include Package 2260
- Hyperlink
  - Action Element 1670
  - As Sub Activities 1670
  - Diagrams 1670
  - Element 1668
  - In Linked Document 931
  - Insert In RTF Template 2291
  - Insert New From Toolbar 143
  - To Element From Linked Document 931
  - To External Files 1669
  - To Help Topics 1668
  - To Internet Facilities 1668
  - To Matrix Profiles 1668
  - To Model Search 1668
  - To Script 1670
  - To Team Review 1668
- | -
- IBM Rational Rapsody
  - Import Model From 420
- Icon
  - Attribute Private 585
  - Attribute Protected 585
  - Checked In Package 585
  - Checked Out Package 585
  - Composite Element 825
  - Hidden Decomposition Indicator 825
  - Namespace Root Package 585
  - Operation Private 585
  - Operation Protected 585
  - Project Browser, User Defined 1308
  - Spectacle, Inside Element 825
  - Version Controlled Package 585
- ICONIX
  - Disable 1925
  - Elements 1925
  - Enable 1925
  - Layout 1925
  - MDG Technology 1925
  - Relationships 1925
  - Roadmap 1925
  - UML Toolbox Pages 1925
- Icons
  - For Element On Diagram 841
  - For Toolbox Items, Assign 1342
- IDE
  - And Enterprise Architect 9
- IDEF1X
  - Data Modeling Notation 1982
- Image
  - Add To MDG Technology 1333
  - Change Linked To Embedded 2345
  - Element, Add To MDG Technology 1356
  - Handling, RTF 2345
  - Manager 755
  - Print Scaled 766
  - Refresh 755
  - Scale To Page Size 766
  - Select Alternative 755
  - Store In Enterprise Architect 755
  - Use From Image Library 758
- Image Element 1672
- Image Library
  - Import 758
  - Use 758
- Image Manager
  - Create Custom Diagram Background 757
  - Select Alternative Image For Element 755
  - Use 755
- IME
  - Code Editor 1805
- Implementation Manager
  - And Enterprise Architect 81
  - Project Role 81
- Implementation Report
  - Generate 2341
  - In Traceability 636
  - Target Types 2342
- Implemented Interfaces
  - Generate/Disable Methods For 1899
- Implements
  - Connector 1254
  - Relationship 1254
- Import
  - .EAB File 372
  - ActionScript, Reverse Engineering 1793
  - ArcGIS XML Workspace 1641
  - Binary Module (Tools Menu Option) 116
  - Binary Module, Languages Supported 1797
  - Binary Module, Reverse Engineering 1797
  - C#, Reverse Engineering 1793
  - C, Reverse Engineering 1793
  - C++, Reverse Engineering 1793
  - Code Files (Tools Menu Option) 116
  - Code Generation Templates 1401
  - Code Templates 1462

## Import

Code, Select Language 142  
 Component Types 1894  
 Data 308  
 Database Schema from ODBC 1973  
 DDL Schema from ODBC 1973  
 Delphi, Reverse Engineering 1793  
 Directory Structure, Reverse Engineering 1795  
 EMX files 419  
 File, Field Substitution Macros 1428  
 From IBM Rational Rhapsody 420  
 From XMI 417  
 Global Elements 2053  
 Handle Classes Not Found 1798  
 Image Library 758  
 Java, Reverse Engineering 1793  
 MDA-Style Transformation Templates 1401  
 MDG Technologies To APPDATA 1288  
 MOF From XMI 2097  
 OWL/RDF .XML Files 1652  
 Pattern 1273  
 PHP, Reverse Engineering 1793  
 Python, Reverse Engineering 1793  
 Reference Data, Automatically 308  
 Reference Data, Introduction 305  
 Reference Data, Manually 308  
 Referenced XML Schema 2053  
 Requirements Via CSV 1481  
 RTF Template 2270, 2316  
 Scenario as Test 2224  
 Scenarios From Package 2224  
 Source Code, Reverse Engineering 1792, 1798  
 Source Directory (Tools Menu Option) 116  
 Split WSDL Files 2086  
 Technology Files (Option) 115  
 Test From Other Element 2226  
 UML Pattern 1273  
 UML Profiles 1279  
 UML2 files 419  
 User ID From Active Directory 257  
 Version Controlled Model Branch 372  
 Visual Basic, Reverse Engineering 1793  
 Visual Basic.Net, Reverse Engineering 1793  
 WSDL Files 2086  
 XMI 431  
 XSD 2053

Import/Export  
 Submenu (Project Menu) 102

Import/Export Submenu  
 Package Context Menu, Project Browser 577

Imported Class Elements 1976  
 in the JBOSS server 1849

in the Options dialog 1922  
 In the Project Browser, right-click on the package into which to paste the copied elements, and select 808  
 in the Scripts tab of the Output win 2378  
 Inbuilt Stereotypes 1663  
 Include  
 Connector 1224  
 Package In Generated Report 2260  
 Relationship 1224  
 including AspectJ extensions 1756  
 Increase Text Display Size 132  
 Indentation  
 Automatic, Common Code Editor 1810  
 Manual, Common Code Editor 1810

Index  
 Data Modeling, Create 1963  
 Data Modeling, What Is A? 1963

Indicator  
 Locked Element 276  
 Visibility 731

Information Engineering  
 Data Modeling Notation 1982

Information Flow  
 And Patterns 1226  
 Connector 1225, 1226  
 In Combination 1226  
 Realized 1229  
 Relationship 1225, 1226

Information Item  
 Conveyed 1228  
 Element 1194  
 Find In Project Browser 938, 1226

Inheritance  
 Connector 1224  
 Relationship 1224

Inherited Attributes  
 Display 895

Inherited Feature  
 Show 745

Inherited Operation  
 Display 908

Inherited Port  
 Manage 1203

Initial  
 Element 1136

Initial Code  
 Operations Properties 906

Inline Features  
 Submenu (Element Menu) 110

Inline IME  
 Option In RTF Editor 2297

Inline Sequence Elements

- Inline Sequence Elements
  - Part And Port 1084
- InnoDB
  - BaseModel Script 224
- In-place Editor
  - Automatic Scroll Into View 916
  - Connector Options 957
  - Edit Attribute Keyword 921
  - Edit Attribute Name 918
  - Edit Attribute Scope 920
  - Edit Attribute Stereotype 919
  - Edit Connector Labels 957
  - Edit Element Name 918
  - Edit Operation Name 918
  - Edit Operation Parameter Keyword 922
  - Edit Operation Parameter Kind 923
  - Edit Operation Scope 920
  - Edit Operation Stereotype 919
  - Element Item, Tasks 916
  - Element Options 916
  - Insert Attribute To Element 924
  - Insert Maintenance Feature 925
  - Insert Operation Parameter 924
  - Insert Operation To Element 924
  - Insert Testing Feature 926
- In-Place Formatting
  - Elements 841
- Input Method Editor
  - Code Editor 1805
- Insert
  - Attribute To Element, In-place Editor 924
  - Bookmarked RTF Into Word 2346
  - Boundary Element 1171
  - Diagram Properties Note 747
  - Linked Elements 833
  - Maintenance Feature In Element 925
  - Operation To Element, In-place Editor 924
  - Related Elements 833
  - Testing Features In Element 926
- Install
  - Enterprise Architect 36
- Instance
  - Define Behavior On Creation, Supported Attributes 1313
  - Paste Object As 733
- Instantiated Template 1183
- Integrated Development Environment 1824
- Integration Testing
  - Display Details 2216
- Integrity
  - Of Model Data 535
  - Of Project Data 535
- Integrity Check 535
- Intellisense
  - Code Editor, Common 1812
  - In Script Editor 2378
  - Keystrokes, Script Editor 2378
- Interaction
  - Behavioral Aspects 910
  - Diagram 1051, 1084, 1087
  - Edit Parameters 910
  - Element 1137
  - Elements and Connectors 710
  - Group, Toolbox 710
  - Parameter 910
  - Parameters, Define 913
- Interaction Diagram
  - Description 1074
  - Diagram 1074
  - Elements And Connectors 1074
  - Example 1076
  - Generate Code From 1778, 1786
- Interaction Occurrence
  - Behavior Call 910
  - Element 1139
- Interaction Operator
  - Combined Fragment 1113
- Interaction Overview Diagram
  - Description 1087
  - Elements And Connectors 1087
  - Example 1088
- InteractionUse
  - Element 1139
- Interface
  - Element 1194
  - Expose Element 1193
  - Provided 1193
  - Required 1193
  - Set For Element 801
  - Source Code Generation 1768
- Intermediary Language
  - Debug, MDA Transformations 1714
  - MDA-Style Transforms 1714
- Intermediate Event, BPEL 1.1
  - Create 1578
  - Model 1578
  - Types 1578
- Intermediate Event, BPEL 2.0
  - Create 1602
  - Model 1602
  - Types 1602
- Internal Binding
  - PIM to PSM 1679
- Internal Editor

- Internal Editor
  - Language Properties 1895
- Internal Mail
  - Context Menu 487
  - Create Message 492
  - Delete Messages 487
  - Forward Message 492
  - Hyperlinks 492
  - Internal Mail 487
  - Model Mail Tab 487
  - Options 487
  - Personal Information 492
  - Personal Information Window 487
  - Reply To Message 492
  - Sent Mail Tab 487
  - Toolbar 487
- Internal Requirement
  - Define 1486
  - Properties 1486
- Internet Browser Applets
  - Java, Debug 1845
- Internet Facilities
  - Hyperlink To 1668
- Internet Search Engine
  - Access Within Enterprise Architect 165
  - Define Default 544
- Interrupt Flow
  - Connector 1230
  - Relationship 1230
- Interruptible Activity Region
  - Element 1141
- Interval Bar
  - Context Menu 1066
  - Timing Diagram Time Interval 1066
- Introduction
  - Copyright Notice 27
  - Enterprise Architect Key Features 14
  - Help File Formats 41
  - License Agreement 28
  - Order Enterprise Architect 35
  - Support 42
  - To Code Engineering 1733
  - To Enterprise Architect 3, 5
  - To Enterprise Architect SDK 1292
  - To Forward Engineering 1733
  - To Quick Linker 791, 1317
  - To Reverse Engineering 1733
  - To Round-trip Engineering 1733
  - To Shape Scripts 1358
  - To Synchronization 1733
  - To Tagged Value Types 1385
  - To UML Objects 1090
- Trademarks 31
- Invocation
  - Associate With Behaviors 911
  - Re-associate With Behaviors 911
  - Synchronize Arguments With Behavior Parameters 911
- Invocation Actions 1787
- Invocations
  - Modeling 910
- Invoke
  - Connector 707
  - Method, Object Workbench 2179
- Issue
  - Create From Test Item 2228
  - Elements And Requirements 1491
- Issue (Defect)
  - Add 2244
  - Automation Interface, Element Package 2492
  - Element 2244
  - Hide Stereotype Letter 2244
  - Show Stereotype Letter 2244
- Issue (Project)
  - Add 463
  - Delete 463
  - Model 462
  - Modify 463
  - Report, Generate 462
- Issues Report
  - Sample Output 466
  - Via Project Issues Tab 466
- isUnique
  - UML Property 964
- Iterate Through EAP File
  - Automation Interface Code Example 2570

## - J -

- Java 1861
  - Advanced Debug Techniques 1845
  - Applets In Internet Browsers, Debug 1845
  - AspectJ Extensions 1793
  - Code Generated From State Machine Diagram 1781
  - Code Generation 1915
  - Debug 1843
  - Debug Session, Attach To VM 1845
  - Debug, System Requirements 1837
  - General Debug Setup 1843
  - Import, Reverse Engineering 1793
  - Language Options 1915
  - Modeling Conventions 1756

Java 1861  
 Modeling Conventions, AspectJ Extensions 1757  
 Sample Model, Visual Execution Analysis 2145  
 Transformation, MDA-Style Transform 1699  
 Web Servers, Debugging 1846

Java Program  
 Application Pattern 1738

JavaScript 2372, 2373

JBOSS  
 Server Configuration 1848  
 Server, Debugging 1846

Jet 3.5  
 Switch To Jet 4.0 47, 544

Jet 4.0  
 Supporting Spell Checking Languages 481  
 Switch From Jet 3.5 47, 544

Join  
 Element 1130, 1133  
 Pseudo-State 1130, 1133

JScript 2372, 2373

Jump  
 Connectors 948

Junction  
 Element 1142

JUnit Transformation  
 MDA-Style Transform 1702

## - K -

Key  
 Combinations 166  
 Database 1951  
 Foreign 1951  
 Missing Combination 153  
 Primary 1951  
 Primary, Create 1953

Key Bindings  
 Code Editor, Common 1819

Key Features  
 Of Enterprise Architect 14

Key Store  
 File Based 2739  
 Network Based 2739

Keyboard  
 Accelerator Map 166  
 And Mouse Shortcuts 174  
 Missing Combination 153  
 Shortcuts 166  
 Shortcuts, Customize 160  
 Shortcuts, Reset 160

Key-Mouse

Combinations 174

Keys  
 Foreign, Definition 1625  
 Primary, Definition 1625

Keystore  
 Troubleshooting 2742

Keywords  
 For RTF Report (Legacy) 2326

## - L -

Label  
 Alignment 760  
 Bold 760  
 Connector 760  
 Connector, Hide/Show 941  
 Direction Indicator 760  
 Display Under Toolbar Icons 155  
 Element 760  
 Hide 760, 957  
 Menu 760  
 ODM Element, Define New 1652  
 Set Color 760  
 Show 957  
 Show And Hide In Toolbox 701  
 Visibility 957

Label Visibility  
 On Sequence Messages 1083

Language  
 Adjust in RTF (Legacy) 2326  
 Adjust in RTF Report 2269  
 Create C# Automatic Properties 893  
 Create Properties As Attributes 892  
 Custom, Create Templates For In Code Template Editor 1466  
 For Spell Check 481  
 Macros 1902  
 Other Than English, Spell Checking 481

Language Options  
 ActionScript 1909  
 Ada 2005 1909  
 ArcGIS 1910  
 C 1911  
 C# 1912  
 C++ 1912  
 Code Editor 1896  
 Code Engineering 1907  
 Code Generation 1907  
 Delphi 1913  
 Java 1915  
 MDG Technology 1921  
 PHP 1916

- Language Options
  - Python 1917
  - Reverse Engineering 1907
  - SystemC 1917
  - VB.NET 1918
  - Verilog 1919
  - VHDL 1919
  - Visual Basic 1920
- Languages
  - Code Generation, Hardware Definition 1741
  - Code Generation, Software 1741
  - Reverse Engineering, Supported 1790
  - Supported By Code Editor 1800
  - Supported Hardware Description Languages 1784
  - Supported, Debugging 1734
  - Supported, Execution Analysis 1734
  - Supported, Recording 1734
- Layout
  - Diagram, Automatically 787
  - Diagram, Move Sections 763
  - ICONIX 1925
  - Sequence Diagram 1077
- Layout Diagram
  - Auto Route Layout 786
  - Box Layout 775
  - Center Focussed Layout 770
  - Circular Layout 770
  - Converge Layout 784
  - Digraph Layout 778
  - Diverge Layout 784
  - Elliptical Layout 770
  - Fan Relations Layout 783
  - Main Procedure 769
  - Neaten Layout 781
  - Options 769
  - Per Page Layout 776
  - Spring Layout 780
  - Tool 769
  - Top-To-Bottom Circle 770
- Layout Tools Window 769
- Lazy Loading
  - Enterprise Architect Performance 210, 212, 214, 216, 218, 219, 221
  - For ASA Data Repository 219
  - For MSDE Server Data Repository 221
  - For MySQL Data Repository 210
  - For Oracle Data Repository 214, 216
  - For PostgreSQL Data Repository 218
  - For Progress OpenEdge Data Repository 221
  - For SQL Server Data Repository 212
- Learning Center
  - Add To MDG Technology 1330
  - Getting Started 89
  - Introduction 40
  - Profiles, Create 1350
  - Run Add-In Functions From 1353
  - Save Profile 1354
  - Tools 89
  - Window 89
- Learning Center Profile
  - Commands, Built In 1352
  - Links To Document Files 1351
  - Menu Commands, Built In 1352
  - Toolboxes, Define 1350
  - Toolboxes, Link To Commands 1350
  - Toolboxes, Link To Document Files 1350
  - Toolboxes, Link To Help Pages 1350
  - Toolboxes, Link To Screens 1350
  - Toolboxes, Link To Video Files 1350
  - Toolboxes, Link To Web Pages 1350
  - View Commands, Built In 1352
- Legend
  - Add To Diagram 748
  - Add To State Machine Table 1046
  - Edit 748
  - Element 748
  - Ignore 1046
  - Insert New From Toolbar 143
  - Never Happen 1046
  - Remove From State Machine Table 1046
  - Style 748
- Letter A
  - Red 928, 929
- Level Numbering
  - For Requirements 1483
  - Show In Project Browser 571
- License
  - Agreement 28
  - Information 2738
  - Management 2736
- Lifecycle
  - Of A Sequence Element 1076
- Lifeline
  - Element 1144
  - Objects In Sequence Diagrams 1079
  - Sequence Element, Termination 1076
- Limit Auto Recording to Stack Frame Threshold
  - Marker Types, Record Sequence Diagrams 2155
- Limitations
  - Of XMI 421
- Line
  - Angles, Tidy 948



- Line
  - Bend At Cursor 948
  - Bezier 948
  - Jumps 948
  - Straighten At Cursor 948
  - Style 948
  - Suppress Segments 948
- Line Points
  - Add 948
  - Delete 948
  - Toggle 948
- Line Selection
  - Code Editor, Common 1810
- Link
  - Add Note 946
  - Copy Between Instances Of Elements From Project Browser 952
  - Create Between Elements 952
  - Create From Project Browser 952
  - Display Options 559
  - Element Feature To Note 815
  - Field Substitution Macros 1429
  - Note To Element Feature 815
  - Paste Object As 733
  - To Element Feature 944
  - Window 648
- Link Notes
  - To Element Feature 815
  - To Internal Documentation 815
- Linked Diagram
  - Composite Element 825
- Linked Document
  - Create 928
  - Create Diagram From 932
  - Create Document Artifact 929
  - Create Element From 932
  - Create For UML Element 929
  - Create Glossary Definition In 930
  - Delete 933
  - Edit 930
  - Editor Context Menu 930
  - Hyperlink 931
  - Introduction 928
  - Link Into RTF Report 928, 2309
  - Replace 933
  - To UML Element 929
- Linked Document Template
  - Add To MDG Technology 1335
  - Assign To Group 934
  - Create 934
  - Delete 934
  - Edit 934
  - Modify 934
- Linked Element
  - Convert To Local Copy 823
- Linked File
  - Field Substitution Macros 1430
- Linked Image
  - Change To Embedded 2345
- List Macro
  - Code Template Syntax 1444
- List Overrides
  - RTF Reports 2299
- Literal Text
  - Code Template Syntax 1406
- Load
  - Controlled Package 429
  - Report Templates (Legacy) 2323
- Loaded Modules
  - Show In Debugger 1887
- Local
  - Directories 1901
  - Path Dialog 1901
  - Paths 1900
  - Pre/Post Conditions 1093
- Local Variables
  - View 1877
  - View Long Values 1879
- Locals Window 1877
  - View Long Values 1879
- Locate Compiler Error
  - In Code, For Build 1866
- Lock
  - Apply User Lock 274
  - Connector 268
  - Delete 268
  - Delete, User level 278
  - Diagram, Rigorous Security Mode 274
  - Diagram, Security Off 767
  - Diagram, User/Group Lock 271
  - Element 268
  - Element, Security Off 837
  - Identify Owner 277
  - Manage 268
  - Manage, User-Level 278
  - Model Elements, Rigorous Security Mode 274
  - Model Elements, User/Group Lock 271
  - Package, Rigorous Security Mode 274
  - Package, User/Group Lock 271
  - Packages 273
  - Release User Lock 274
  - Standard Security Policy 271
  - View 268
  - View, User Level 278

- Locked Element
    - Add Connectors 273
    - Indicators 276
  - Log
    - Audit, Clear 384
    - Audit, Load 384
    - Audit, Save 384
  - Logical Diagram
    - Class Diagram 1014
  - Logical Model 668
    - Template (Class) 662
  - Login
    - As Different User (Security) 104
    - Group 262
    - Multiple Under One ID 262
  - Loop Node
    - Structured Activity 1161, 1164
- M -**
- Macro
    - Behavioral Model 1449
    - Branching 1446
    - Code Template Syntax 1409, 1444
    - Control 1444
    - CONVERT\_DB\_TYPE 1728
    - CONVERT\_NAMES 1728
    - CONVERT\_TYPE 1728
    - Direct Substitution 1406
    - EASL Code Generation 1449
    - EASL\_GET 1449
    - EASList 1449
    - Field Substitution, Code Template Syntax 1411
    - Function, Code Template Syntax 1439
    - Language 1902
    - List 1444
    - PI 1444, 1448
    - Preprocessor 1902
    - Processing Instruction 1448
    - REMOVE\_PREFIX 1728
    - SQL Search 623
    - Synchronization 1447
    - Tagged Value, Code Template Syntax 1437
    - Template Substitution, Code Template Syntax 1411
    - TRANSFORM\_CLASSIFIER 1730
    - TRANSFORM\_CURRENT 1724, 1727
    - TRANSFORM\_REFERENCE 1724, 1730
    - TRANSFORM\_TAGS 1727
  - Mail Interface Package
    - Automation Interface 2566
    - MailInterface, Automation Interface 2566
  - MailInterface Class
    - Automation Interface, Mail Interface Package 2566
  - Main Context Menu 90
  - Main Menu 90
  - Maintain
    - Groups 262
    - Security Users 256
  - Maintenance 2238
    - Asterisk On Maintenance Window Tabs 2237
    - Compartment, Element 842
    - Create Defect Item From Test 2228
    - Dialog 1001
    - Elements And Connectors 717
    - Feature, Insert In Element 925
    - Group, Toolbox 717
    - Items 2236
    - Model Template 662, 673
    - Of Element Properties 2238
    - Problem Types 1001
    - Script, Show In Compartments 2242
    - Support 2238
    - Testing Types 1002
    - Window 2237
    - Workspace 2237
  - Maintenance Data
    - Project Types Menu Option 129
  - Maintenance Diagram
    - Description 2234
    - Elements And Connectors 2234
    - Example 2235
  - Maintenance Item
    - Copy Between Categories 2240
    - Create Element From 2241
    - Move Between Categories 2240
  - Make Same
    - Submenu 112
  - Manage
    - Add-Ins 2593
    - Baselines 398
    - Bookmarks 530
    - Changes 311
    - Diagrams 721
    - Elements 795
    - Inherited Ports 1203
    - Locks 268
    - MDG Technologies 1285
    - Redefined Ports 1203
    - Requirements 1488
    - User-Level Locks 278
    - Views 676
    - Views, Add Views 677

- Manage
  - Views, Delete 678
  - Views, Rename 678
- Manage .EAP File
  - Submenu (Data Management Menu) 120
- Manage Uses Context List Dialog 861
- Managed C++
  - Modeling Conventions 1752
- Manifest
  - Connector 1231
  - Relationship 1231
- Manual Version Control
  - With XMI 432
- Map
  - Data Types Between DBMSs 1950
- Map State Changes
  - Visual Execution Analyzer 2168
- Mapper
  - Data Type Conversion, Table 1946
- Marker
  - Record, Set 2155
  - Sets 2160
  - Types 2155, 2160
- Marker Management
  - Debugger 1869
- Marker Sets
  - Create 2158
  - Delete 2158
  - Maintain 2158
  - Work With 2158
- Masked Tagged Values
  - Create 1393
- Master Document
  - Element 2327
  - Element, Create 2330
  - Generate Documentation 2338
  - Generate HTML Documentation 2368
  - Overview, Virtual Document 2327
  - RTF Bookmarks In 2327
  - Tagged Values 2330
- Master Document (HTML) 2368
- Matrix
  - Swimlanes 753
- Matrix Profile
  - Hyperlink To 1668
- MDA Transform
  - Add To MDG Technology 1333
- MDA Transformation
  - Built-In 1679
  - Debug Intermediary Language 1714
  - Enterprise Architect 9
  - Overview 1679
  - Tools Menu Option 118
- MDA-Style Transformation
  - Built In Transformation 1686
  - C# Transformation 1687
  - Chaining Transformations 1684
  - Convert Datatypes 1728
  - Convert Names 1728
  - Convert Types 1728
  - Copy Information 1727
  - Copy Tagged Values 1727
  - Create Connectors 1722
  - Cross References 1730
  - Data Model To ERD Transformation 1689
  - DDL Transformation 1689
  - Duplication Of Connectors 1724
  - EJB Transformations 1694
  - ERD To Data Model Transformation 1697
  - Export Transformations 1400
  - Import Transformation Templeness 1401
  - Intermediary Language 1714
  - Java Transformation 1699
  - JUnit Transformation 1702
  - NUnit Transformation 1704
  - Remove Prefix 1728
  - Specify Classifiers 1730
  - Transform Connectors 1724
  - Transform Elements 1683
  - Transformation Templates 1710
  - Write Transformations 1712
  - WSDL Transformation 1705
  - XSD Transformation 1707
- MDDE
  - Advanced Debug Techniques, Java 1845
  - Analyzer Scripts, Manage 1828
  - Available Tools 1826
  - Basic Setup 1827
  - Breakpoint Management 1869
  - Build Script, Create 1832
  - Build Script, Introduction 1832
  - Debug .NET 1850
  - Debug .NET CLR Versions 1851
  - Debug .NET With COM Interop Process 1852
  - Debug Apache Tomcat Server Configuration 1849
  - Debug Apache Tomcat Windows Service 1850
  - Debug ASP .NET 1853
  - Debug Java 1843
  - Debug Java Applets In Internet Browsers 1845
  - Debug Java Web Servers 1846
  - Debug JBOSS Server Configuration 1848
  - Debug Symbols, C++ And Native Applications 1842

## MDDE

- Debugger Frameworks 1867
- Debugger System Requirements 1837
- Debugger, Overview 1867
- External Tools 1827
- For C++ Applications 1841
- For Microsoft Native Applications 1841
- For WINE Applications 1839
- General Debug Setup, Java 1843
- General Workflow 1825
- Generate Code 1825
- Getting Started 1825
- Introduction 1824
- Java Debug Session, Attach To VM 1845
- Limitations 1734
- Marker Management 1869
- Overview 1734
- Prerequisites 1825
- Recursive Builds 1832
- Script Actions, Define 1831
- Set Up Debug Session 1837
- Set Up Debug Session For .NET 1850
- Supported Environments 1734
- Synchronize Code 1825
- UAC-Enabled Operating Systems 1838
- Workspace Layout 1826

## MDG Add-In For

- CORBA 1289
- Data Distribution Service 1289
- DDS 1289
- Department Of Defense Architecture Framework -  
Ministry Of Defence Architecture Framework  
1289
- DoDAF-MODAF 1289
- DOORS 1289
- Eclipse 1289

## MDG Add-Ins

- Add-In Model 2671
- MDG Events 2672
- MDG\_BuildProject 2673
- MDG\_Connect 2674
- MDG\_Disconnect 2675
- MDG\_GetConnectedPackages 2676
- MDG\_GetProperty 2676
- MDG\_Merge 2678
- MDG\_NewClass 2680
- MDG\_PostGenerate 2681
- MDG\_PostMerge 2682
- MDG\_PreGenerate 2683
- MDG\_PreMerge 2683
- MDG\_PreReverse 2684
- MDG\_Run\_Exe 2685

MDG\_View 2686

## MDG Events

- Add-In Model 2672
- MDG\_BuildProject 2673
- MDG\_Connect 2674
- MDG\_Disconnect 2675
- MDG\_GetConnectedPackages 2676
- MDG\_GetProperty 2676
- MDG\_Merge 2678
- MDG\_NewClass 2680
- MDG\_PostGenerate 2681
- MDG\_PostMerge 2682
- MDG\_PreGenerate 2683
- MDG\_PreMerge 2683
- MDG\_PreReverse 2684
- MDG\_Run\_Exe 2685
- MDG\_View 2686

## MDG Integration For

- Eclipse 1289
- TcSE 1289
- Teamcenter Systems Engineering, Siemens PLM  
1289
- Visual Studio 2005/2008 1289

## MDG Link For

- DOORS, Telelogic 1289
- Eclipse 1289
- Visio 1289
- Visual Studio.NET 1289

## MDG Technology

- Access, Remote From Enterprise Architect  
1287
- Activate 1285
- Active, Automatically Pinned Pages 701
- And Resources Window 1284
- Code Generation 1921
- Create 1324
- Create Toolbox Profile For 1337
- Define Learning Center Profile 1350
- Define Validation Configuration 1354
- Deploy From Add-In 1357
- Deploy From File 1357
- Disable 1285
- Enable 1285
- Example Of Development 1323
- Example Of Element Image Setup 1356
- Import To APPDATA 1288
- Include Custom Diagram Types 1346
- Incorporate Model Template 1355
- Introduction 1323
- Introduction, Technology Sources 1283
- Language Options 1921
- Link for Downloads 1283

- MDG Technology
  - Manage 1285
  - ODM 1643
  - Ontology Definition Metamodel 1643
  - Toolbox Groups 1284
  - Win32® User Interface 1661
- MDG Technology (Integrated)
  - ArchiMate 1621
  - BPEL 1570
  - BPMN 1545
  - Data Flow Diagram 1506
  - Entity Relationship Diagram 1627
  - Eriksson-Penker 1623
  - Gang Of Four Pattern 1928
  - GoF Pattern 1928
  - ICONIX 1925
  - Mind Mapping 1503
  - SoaML 2088
  - SOMF 2092
  - SPEM 523
  - SysML 1990
- MDG Technology For
  - CORBA 1289
  - Data Distribution Service 1289
  - DDS 1289
  - Department Of Defense Architecture Framework - Ministry Of Defence Architecture Framework 1289
  - DoDAF-MODAF 1289
  - Enterprise Java Beans 1289
  - Python 1289
  - SysML 1289
  - Systems Modeling Languages 1289
  - Testing 1289
  - The Open Group Architecture Framework 1289
  - TOGAF 1289
  - Zachman Framework 1289
- MDG Technology Selection (MTS) File 1335
- MDG Technology Wizard
  - Add Code Modules 1331
  - Add Diagram Type To Technology 1329
  - Add Images 1333
  - Add Learning Center To Technology 1330
  - Add Linked Document Template To Technology 1335
  - Add MDA Transforms 1333
  - Add Pattern To Technology 1328
  - Add Profile To Technology 1327
  - Add RTF Report Template To Technology 1334
  - Add Scripts 1334
  - Add Tagged Value Types 1331
  - Add Toolbox To Technology 1329
  - Create Technologies 1324
- MDG\_BuildProject
  - Add-In Model 2673
- MDG\_Connect
  - Add-In Model 2674
- MDG\_Disconnect
  - Add-In Model 2675
- MDG\_GetConnectedPackages
  - Add-In Model 2676
- MDG\_GetProperty
  - Add-In Model 2676
- MDG\_Merge
  - Add-In Model 2678
- MDG\_NewClass
  - Add-In Model 2680
- MDG\_PostGenerate
  - Add-In Model 2681
- MDG\_PostMerge
  - Add-In Model 2682
- MDG\_PreGenerate
  - Add-In Model 2683
- MDG\_PreMerge
  - Add-In Model 2683
- MDG\_PreReverse
  - Add-In Model 2684
- MDG\_Run\_Exec
  - Add-In Model 2685
- MDG\_View
  - Add-In Model 2686
- MDGMenus Enum
  - Automation Interface 2418
- Memory Viewer
  - Window 1886
- Menu
  - Add-Ins 125
  - Analyzer 120
  - Context 86
  - Context, Project Browser 569
  - Customize Appearance 161
  - Delete Options 161
  - Diagram 105
  - Diagram Context 687
  - Diagram, Advanced 106
  - Edit 93
  - Element 108
  - Element Context 827
  - File 92
  - Help 131
  - Items, Define In Add-In 2588
  - Label 760
  - Main 86, 90
  - Main Context 90

- Menu
  - Missing Option 153
  - More Element Tools 95
  - More Project Tools 95
  - New Element Or Connector Context 690
  - Package Control 424
  - Project 100
  - Properties, Element Context 828
  - Record & Analyze Context 2151
  - Set Animation 161
  - Set Shadowing 161
  - Settings (Project) 127
  - Toolbars (View Menu Option) 95
  - Tools 115
  - Version Control (Package) 357
  - Version Control (Project) 355
  - View 95
  - Visual Styles (View Menu Option) 95
  - Window 129
- Menu Option (File Menu)
  - Close Project 92
  - Exit 92
  - New Project 92
  - Open Project 92
  - Page Setup 92
  - Print 92
  - Print Preview 92
  - Print Setup 92
  - Reload Current Project 92
  - Save Project As 92
- Merge
  - Baseline With Current Model, Overview 394
  - Element 1117, 1145
  - Node 1145
  - Options, Compare Utility 408
- Merge Packages
  - Relationship 1253
- Message 959
  - Asynchronous Signal 1241
  - Collaboration 1244
  - Colors In Communication Diagrams 1086
  - Communication 1242
  - Communication, Coloring 561
  - Communication, Create 1243
  - Connector 1231
  - Create On Timing Diagram 1246
  - Endpoint 1145
  - Group, Start New 1244
  - Label 1146
  - Level 1244
  - Move 1231
  - Recursion 1255
  - Relationship 1231
  - Re-Order 1244
  - Scope 959
  - Self Message 1235
  - Self Message Call 1236
  - Sequence Communication 1244
  - Sequence Diagram, Asynchronous Signal 1241
  - Sequence Diagram, Examples 1237
  - Sequence Diagram, General Ordering 1240
  - Sequence Diagram, Self Message 1235
  - Sequence, Create 1232
  - Sequence, Label Visibility 1083
  - Sequencing 1244
  - Source and Target 959
  - Timing Diagram 1245
  - WSDL Diagram 2073
  - WSDL Element 2073
- Message Angle
  - Adjust With Duration Constraint 1238
- Message Part
  - WSDL Attribute 2074
- MessageFlag Enum
  - Automation Interface 2419
- Meta Object Facility
  - Introduction 2094
- Metaclass
  - Add To Profile 1297
- META-INF Package 1694
- Metamodel
  - Elements and Connectors 715
  - Group, Toolbox 715
- Metasequence
  - Code Editor Find And Replace 1814
  - Helper Menu 1814
- Method
  - Add And Delete, Automation Interface Code Example 2575
  - Automation Interface, ElementFeatures Package 2400
  - Context Menu, Project Browser 582
  - Create Use Case For 1807
  - Delete If Not In Code In Reverse Synchronization 1899
  - From Rule Flow Activity 1533
  - Implemented Interfaces 1899
  - Include Bodies In Model When Reverse Engineering 1899
  - Invoke, Object Workbench 2179
  - Link To Use Case 1807
  - Show Parameters On Diagram 732
  - Work With, Automation Interface Code Example 2582

- MethodConstraint
  - Automation Interface, ElementFeatures Package 2404
- Methodology
  - FDD 1485
  - Feature-Driven Design 1485
- MethodTag
  - Automation Interface, ElementFeatures Package 2405
- Metric
  - Automation Interface, Element Package 2494
  - Field Substitution Macros 1431
- Metric Types 457
  - Define 457
  - Global 457
  - Non-Global 455
- Metrics 1000
  - Project Management Window 448
- Metrics And Estimation
  - Default Hour Rate 520
  - ECF 518
  - Effort Types 456
  - Environment Complexity Factors 518
  - For An Element 455
  - Metric Types 457
  - Risk Types 458
  - TCF 517
  - Technical Complexity Factors 517
- MFC Windows
  - Application Pattern 1738
- Microsoft .NET 1850
- Microsoft C++
  - Sample Model, Visual Execution Analysis 2145
- Microsoft Native
  - Debug Symbols 1842
  - Set Up Debug Sessions 1841
- Microsoft Word
  - Special Considerations 2347
  - Use In RTF Documentation 2344
- Microsoft.NET
  - Sample Model, Visual Execution Analysis 2145
- Migrate
  - BPMN 1.0 Model To 1.1 1565
  - BPMN 1.1 Model To 2.0 1567
  - SysML 1.1 Model To 1.2 2024
- Migrate()
  - Function, BPMN 1565
  - Function, BPMN 2.0 1567
  - Function, SysML 2024
- MigrateToBPMN11()
  - Function 1565, 2535
- Migration
  - From UML 1.3 535
  - To UML 2.0 535
- Mind Mapping
  - Concept 1503
  - Diagram 1503
  - Disable 1503
  - Elements 1503
  - Enable 1503
  - MDG Technology 1503
  - Relationship 1503
  - Toolbox Page 1503
- MiscData 2478
- Model
  - Activity, BPEL 1.1 1581
  - Activity, BPEL 2.0 1605
  - Add To Project 178
  - Analysis 1503
  - Analysis, Business Processes 1517
  - Automation Interface 2390
  - BPEL 1.1 1571
  - BPEL 1.1 Process 1576
  - BPEL 1.1, Create 1573
  - BPEL 2.0 1595
  - BPEL 2.0 Assign Activity 1613
  - BPEL 2.0 Process 1599
  - BPEL 2.0, Create 1597
  - BPMN 2.0, Serialize In XML 1569
  - Business 1512
  - Business Domain, Create 1529
  - Business Processes, Analysis 1517
  - Business Rules 1522
  - Contents 660
  - Context Menu, Project Browser 569
  - Create Using Model Wizard 660
  - Data Integrity 535
  - Data Object, BPEL 2.0 1610
  - Databases 1625
  - Default Diagram, Cancel 744
  - Default Diagram, Set 744
  - Default Fonts 552
  - Delete 569, 660
  - Delete Element From 811
  - Delete Multiple Elements From 811
  - Gateway, BPEL 1.1 1584
  - Gateway, BPEL 2.0 1607
  - Glossary 467
  - Integrity Check 535
  - Integrity, Run SQL Patches 539
  - Issues, Record 462
  - Navigation Tools 566
  - Package, Create Using Model Wizard 660
  - Pattern 660

- Model
- Pool, BPEL 1.1 1588
  - Property, BPEL 2.0 1611
  - Remove Recent 89
  - Requirements 1468, 1483
  - Requirements, SysML 2018
  - Root Node 569
  - Search Tools 566
  - Sequence Flow, BPEL 1.1 1587
  - Sequence Flow, BPEL 2.0 1612
  - Sharing, Introduction 240
  - Shortcut 183
  - Shortcut (Capture Current Environment) 187
  - Shortcut (Direct Definition) 185
  - Simulation 2100
  - Structure 657
  - SysML Operational Domain 2019
  - SysML Parametric, Create 2013
  - SysML System Design 2021
  - SysML, Simulate 2016
  - Systems Engineering, Create 1984
  - Templates, Incorporate In Technology 1355
  - Trace Tools 566
  - Transformation 1683
  - Transformations 1679
  - UML, Specialized 1498
  - Upgrade 537
  - What Is A? 660
  - WSDL 2069
  - WSDL, Binding 2078
  - WSDL, Document 2083
  - WSDL, Message 2073
  - WSDL, Message Part 2074
  - WSDL, Namespace 2067, 2070
  - WSDL, Port Type 2075
  - WSDL, Port Type Operation 2076
  - WSDL, Service 2082
  - XSD 2027
- Model Branch
- .EAB File 371
  - Apply Version Control 371
  - Export 371
  - File 371
  - Import 372
- Model Changes
- Auditing 382
  - Record 382
- Model Document
- Add Packages As Attributes 2333
  - Change Package Sequence 2335
  - Delete Package Attributes 2334
  - Delete Packages 2334
  - Document Order 2335
  - Element 2327
  - Element, Create 2331
  - Generate Documentation 2338
  - Generate HTML Documentation 2368
  - Identify Search For 2331
  - Model Search Sequence 2335
  - Move Package Attributes Between Elements 2335
  - Overview, Virtual Document 2327
  - Select Template For 2331
  - Sequence of Model Document Elements 2335
  - Sequence of Package Attributes 2335
  - Tagged Values 2331
- Model Document (HTML) 2368
- Model Driven Achitecture
- Overview 1679
- Model Driven Development Environment 1824
- Advanced Debug Techniques, Java 1845
  - Analyzer Scripts, Manage 1828
  - Available Tools 1826
  - Basic Setup 1827
  - Breakpoint Management 1869
  - Build Script, Create 1832
  - Build Script, Introduction 1832
  - Debug .NET 1850
  - Debug .NET CLR Versions 1851
  - Debug .NET With COM Interop Process 1852
  - Debug Apache Tomcat Server Configuration 1849
  - Debug Apache Tomcat Windows Service 1850
  - Debug ASP .NET 1853
  - Debug Java 1843
  - Debug Java Applets In Internet Browsers 1845
  - Debug Java Web Servers 1846
  - Debug JBOSS Server Configuration 1848
  - Debug Symbols, C++ And Native Applications 1842
  - Debugger Frameworks 1867
  - Debugger System Requirements 1837
  - Debugger, Overview 1867
  - External Tools 1827
  - For C++ Applications 1841
  - For Microsoft Native Applications 1841
  - For WINE Applications 1839
  - General Debug Setup, Java 1843
  - General Workflow 1825
  - Generate Code 1825
  - Getting Started 1825
  - Java Debug Session, Attach To VM 1845
  - Limitations 1734
  - Marker Management 1869



- Model Driven Development Environment 1824
  - Overview 1734
  - Prerequisites 1825
  - Recursive Builds 1832
  - Script Actions, Define 1831
  - Set Up Debug Session 1837
  - Set Up Debug Session For .NET 1850
  - Supported Environments 1734
  - Synchronize Code 1825
  - UAC-Enabled Operating Systems 1838
  - Workspace Layout 1826
- Model Driven Generation
  - Add-Ins 1289
- Model File
  - Connect To ASA Data Repository 219
  - Connect To Data Repository 209
  - Connect To MSDE Server Data Repository 221
  - Connect To MySQL Data Repository 210
  - Connect To Oracle 10g Data Repository, Via ODBC 214
  - Connect To Oracle 10g Data Repository, Via OLE DB 216
  - Connect To Oracle 11g Data Repository, Via ODBC 214
  - Connect To Oracle 11g Data Repository, Via OLE DB 216
  - Connect To Oracle 9i Data Repository, Via ODBC 214
  - Connect To Oracle 9i Data Repository, Via OLE DB 216
  - Connect To PostgreSQL Data Repository 218
  - Connect To Progress OpenEdge Data Repository 221
  - Connect To SQL Server Data Repository 212
  - Create Access 2007 Repository 194
  - Create Adaptive Server Anywhere Repository 198
  - Create Data Repository 194
  - Create Model 189
  - Create MSDE Server Repository 198
  - Create MySQL Repository 195
  - Create Oracle 10g Server Repository 197
  - Create Oracle 11g Server Repository 197
  - Create Oracle 9i Server Repository 197
  - Create PostgreSQL Repository 197
  - Create Progress OpenEdge Repository 199
  - Create SQL Server Repository 195
  - Enterprise Architect Project Files 178
  - Set Up Adaptive Server Anywhere ODBC Driver 205
  - Set Up Database Model Files 193
  - Set Up MySQL ODBC Driver 201
  - Set Up ODBC Driver 200
  - Set Up Oracle ODBC Driver 208
  - Set Up PostgreSQL ODBC Driver 202
  - Set Up Progress OpenEdge ODBC Driver 206
- Model Glossary
  - Add Item, Glossary Detail Dialog 468
  - Add Item, Glossary Dialog 469
  - Delete Item, Glossary Dialog 469
  - Delete Item, Project Glossary Tab 468
  - Filter List, Glossary Dialog 469
  - Filter List, Project Glossary Tab 468
  - Glossary Dialog 469
  - Glossary Report 471
  - Modify Item, Glossary Detail Dialog 468
  - Modify Item, Glossary Dialog 469
  - Project Glossary Tab 468
  - Redefine Entry Type, Project Glossary Tab 468
- Model Mail
  - Create Message 492
  - Forward Message 492
  - Personal Information Window 487
  - Reply To Message 492
- Model Maintenance (.EAP)
  - Compact Project 539
  - Introduction 534
  - Rename Project 539
  - Repair Project 540
- Model Message
  - Dialog 492
- Model Package
  - Publish, XMI 414
- Model Pattern
  - Introduction 662
  - Select In Enterprise Architect 49
- Model Profile
  - Apply 753
  - Save 753
  - Swimlanes Matrix 753
  - Zachman 753
- Model Search
  - Access From Add-In 2594
  - Access From Shortcut 183, 187
  - Access From Shortcut (Direct Definition) 185
  - Act On Results 614
  - Add Data Columns 611
  - Add Filters 627
  - Advanced Search Options 626
  - Bookmark Result Items 614
  - Conditions 629
  - Context Menu 614
  - Copy Results To Clipboard 614
  - Create Linked Document 614
  - Create Search Definition 623

- Model Search
  - Customize Search View 619
  - Define In MTS File 1335
  - Delete Result Items 614
  - Display Properties Of Result Item 614
  - Export Search Definitions 621
  - Fields 629
  - Filter Bar 619
  - Generate Report On 614
  - Generate RTF Report 2253
  - Group Results 611
  - Hyperlink To 1668
  - Import Search Definitions 621
  - Introduction 611
  - Locate Result Items In Model 614
  - Manage 621
  - Notes Options 611
  - Options 611
  - Print Results 614
  - Process Results Of Search 611
  - Remove Data Columns 611
  - Scripts 614, 2375
  - Sort And Select 619
  - To Populate Model Views 604
  - Toolbar Options 611
  - Value Grouping 619
  - View 611
  - View Header 619
- Model Simulation
  - Activate Simulation Script 2111
  - Breakpoint Sets 2114
  - Breakpoints, Using 2114
  - Edition Support 2102
  - How It Works 2102
  - Multi-Threaded 2137
  - Run Simulation Script 2112
  - Set Up Simulation Script 2109
  - Supported Platforms 2102
  - Topics 2100
- Model State Machine
  - Active State Logic 1784
  - Designate Driving Triggers 1784
  - Establish Port-Trigger Mapping 1784
  - For Hardware Description Languages 1784
- Model Template
  - Business Process 662, 665
  - Class 662, 668
  - Component 662, 670
  - Database 662, 669
  - Deployment 662, 671
  - Domain 662, 667
  - Introduction 662
  - Logical 662, 668
  - Maintenance 662, 673
  - Physical 662, 671
  - Project 674
  - Project Management 662
  - Relational Database 662, 669
  - Requirements 662, 666
  - Testing 662, 672
  - Use Case 662, 666
  - User Interface 662
- Model Transfer
  - Introduction 411
- Model Transformations
  - Submenu (Tools Menu) 118
- Model Validation
  - BPEL 1619
  - Configure 2205
  - Define Configuration For MDG Technology 1354
  - Element Composition Rule 2208
  - Object Constraint Language 2203
  - OCL 2203
  - OCL Conformance Rule 2209
  - Property Validity Rule 2208
  - Rule, Element Composition 2208
  - Rule, OCL Conformance 2209
  - Rule, Property Validity 2208
  - Rule, Well Formedness 2207
  - Rules 2206
  - Submenu (Project Menu) 102
  - Well Formedness Rule 2207
- Model Validation Broadcasts
  - Add-In Model 2621
  - EA\_OnEndValidation 2623
  - EA\_OnInitializeUserRules 2622
  - EA\_OnRunAttributeRule 2627
  - EA\_OnRunConnectorRule 2626
  - EA\_OnRunDiagramRule 2625
  - EA\_OnRunElementRule 2624
  - EA\_OnRunMethodRule 2628
  - EA\_OnRunPackageRule 2625
  - EA\_OnRunParameterRule 2629
  - EA\_OnStartValidation 2623
  - Model Validation Example 2630
- Model Views
  - Automatic Notification 532
  - Context Menu Options 601
  - Define In MTS File 1335
  - Define Search 604
  - Delete (Context Menu) 601
  - Delete (Toolbar) 600
  - Display Recent Postings 604
  - Enable Technology-Defined View 604

- Model Views
  - Export As XML 604
  - Favorites 598
  - Favorites Folder, Create (Context Menu) 601
  - Favorites Folder, Create (Toolbar) 600
  - For Requirement Change Management 1491
  - Import As XML 604
  - Monitor Work Flow 532
  - Move Objects Between Views 604
  - Move Objects Into Favorites 604
  - My Views 598
  - Operations 604
  - Postings From Team Review 598
  - Properties 604
  - Recent Discussions 598
  - Recent Post Options 601
  - Refresh (Toolbar) 600
  - Refresh Search 604
  - Root Node, Create (Context Menu) 601
  - Root Node, Create (Toolbar) 600
  - Slide Show, Run (Manual) 607
  - Slideshow 598
  - Slideshow Folder, Create (Toolbar) 600, 607
  - Slideshow, Automate 607
  - Technology-Defined 598
  - Toolbar Options 600
  - Use Objects In Views And Favorites 604
  - View, Create (Context Menu) 601
  - View, Create (Toolbar) 600
  - Views 598
  - Views Folder, Create (Context Menu) 601
  - Views Folder, Create (Toolbar) 600
  - Window 598
- Model Violations
  - Examples 2203
- Model Wizard
  - Add Model Package 660
  - Quick Start 49
  - Use To Create Model 189
- Modeling
  - Behavior Calls 910
  - Behaviors (General) 899
  - Enterprise Architect Processes 7
  - Enterprise Architect Tools and Features 86
  - Enterprise Architect, Getting Started 44
  - Geodatabases 1633
  - Getting Started - Enterprise Architect 44
  - Invocations 910
  - Systems Engineering 1984
  - Tasks With Enterprise Architect 7
  - The Business Process 1515
  - With Diagrams 687
  - With Elements 795
  - With Enterprise Architect, Overview 659
- Modeling Conventions 1741
  - ActionScript 2 and 3 1742
  - Ada 2005 1743
  - ANSI C 1745
  - C 1745
  - C# 1748
  - C, Object Oriented Programming 1747
  - C++ 1750
  - C++, Managed 1752
  - C++/CLI Extensions 1753
  - Delphi 1754
  - Java 1756
  - Java AspectJ Extensions 1757
  - Object Oriented Programming in C 1747
  - PHP 1758
  - Python 1759
  - SystemC 1760
  - VB.NET 1761
  - Verilog 1763
  - VHDL 1764
  - Visual Basic 1766
- Modeling Language
  - Extending 1292
- Modeling Tool
  - Enterprise Architect 3
- Models Collection 2448
- Model-View-Controller Pattern 1665
- ModelWatcher
  - Automation Interface Repository 2432
- Modify
  - Element Changes 2238
  - Element Defects 2238
  - Element Issues 2238
  - Element Tasks 2238
  - Linked Document Template 934
  - Project Task 461
  - Relationship Using Matrix 644
  - RTF Style Template (Legacy) 2323
  - Tagged Values 972
  - Z-Order 687
- Modules
  - Window 1887
- MOF
  - Export To XMI 2097
  - Getting Started 2095
  - Import From XMI 2097
  - Introduction 2094
- Monitor
  - Events 532
  - Progress 532

- Monitor
    - Work Flow 532
  - Monitor Project Tasks
    - Personal Information Window 496
  - Monitor Workflow
    - Personal Information Window 498
  - More Analyzer Windows
    - Memory Viewer 125
    - Modules 125
    - Scripts 125
    - Submenu 125
    - Workbench 125
  - More Element Tools
    - Submenu 98
  - More Project Tools
    - Submenu 98
  - Mouse/Keyboard
    - Shortcuts 174
  - Mouseovers
    - Code Editor, Common 1812
  - Move
    - Attributes Between Elements 824
    - Connectors 946
    - Connectors, Quick Start 57
    - Diagram Sections 763
    - Diagrams, Quick Start 57
    - Elements Between Packages 806
    - Elements By Increments 112, 763
    - Elements In Diagrams 805
    - Elements, Quick Start 57
    - Internal Responsibility To External Requirement 1486
    - Maintenance Item Between Categories 2240
    - Nudge Elements 112
    - Objects Between Packages 806
    - Operations Between Elements 824
    - Package Contents Up Or Down 586
    - Packages, Quick Start 57
    - Submenu 112
    - Test Between Categories 2223
  - MS Word
    - Update RTF Report Links 2351
    - Use In RTF Documentation 2344
  - MS Word Tables
    - Apply Styles 2349
    - Manipulate 2349
    - Resize 2349
  - MSDE
    - Server Data Repository, Connect To 221
    - Server Repository, Create New 198
    - Upsize To 236
  - MTS File
    - Advanced Options 1335
    - Create 1335
    - Incorporate Model Search 1335
    - Incorporate Model View 1335
    - Working With 1335
  - Multi-page Diagram
    - Print 766
  - Multiple Login
    - Under One User ID 262
  - Multiple Select
    - Items From Project Browser 735
  - Multiple Stereotype
    - Restrict Application Of 1313
  - Multiplicity 662, 667
    - Connector, Source Role 964
    - Connector, Target Role 967
    - Of Connector 986
    - Of Element 986
    - Of Element, Displayable 829
    - Of Element, Non-Displayable 846
  - MVC Pattern 1665
  - MyISAM
    - BaseModel Script 224
  - MySQL
    - Create Repository 195
    - Data Repository, Connect To 210
    - Datatypes For 1949
    - ODBC Driver, Set Up 201
    - Table Type, Set 1939
    - Upsize To 224
- N -**
- Name Template
    - Foreign Key 1960
    - Primary Key 1952
  - Namespace
    - And Whitespace Characters 1777
    - Clear 1777
    - Dialog 1777
    - Explanation 1777
    - List 1777
    - Locate In Project Browser 1777
    - ODM Package, Define New 1652
    - Root 1777
    - Root Package Icon 585
    - Set 1777
    - WSDL Element 2067, 2070
  - Naming Format
    - Camel Case 1728
    - Pascal Case 1728

- Naming Format
    - Spaced 1728
    - Underscored 1728
  - N-Ary
    - Association Element 1673
    - Element (Entity Relationship Diagram) 1627
  - Navigate
    - Diagram 610
  - Navigation And Selection
    - Hotkeys, For Diagram 740
  - Navigation Compass 135
  - Navigation Tools 566
  - Neaten Diagram Layout 781
  - Nested Version Control Packages 316, 320
  - Nesting
    - Connector 1249
    - Relationship 1249
  - New Action Dialog 1095
  - New Code Sections
    - Add To Existing Features 1403
  - New Project
    - Menu Option (File Menu) 92
  - New Search Query 623
  - New Structured Activity Dialog 1161
  - Node
    - Element 1196
  - Non-Selectable
    - Element 837
  - Normal.rtf
    - Template File 2270
  - Notation
    - Co-Region 1242
    - Data Modeling 1982
    - IDEF1X 1982
    - Information Engineering 1982
    - Process Modeling 1518
    - UML 2.1 1982
  - Note
    - Add To Link/Connector 946
    - Attach To Connector 793
    - Create Project Glossary Item 977
    - Element 1147
    - Element, Create 814
    - For Attribute 977
    - For Connector 977
    - For Diagram 977
    - For Element 977
    - For Operation 977
    - Formatting 979
    - Hyperlink Glossary Term 977
    - Insert Glossary Item 977
    - Insert New From Toolbar 143
    - Keyboard Shortcuts 979
    - Link To Element Feature 815
    - Link To Internal Documentation 815
    - Spell Check 977
    - Tab 977
    - Toolbar Options 979
    - Window 977
  - Notelink
    - Connector 1249
    - Insert New From Toolbar 143
    - Relationship 1249
  - Notes
    - Gap, Edit 654
  - Nudge
    - Elements 112, 763
  - Numbered Sections
    - In RTF Reports 2299
  - Numbering Levels
    - Apply 2299
    - In Virtual Documents 2337
    - RTF Reports 2299
    - Use 2299
  - Numbering List
    - RTF Reports 2299
  - Numeric Range Generator
    - Timeline Element States 1062
  - NUnit Transformation
    - MDA-Style Transform 1704
- O -**
- Object
    - Appearance, Options 556
    - Attributes 1716
    - Classes 1716
    - Classifiers 898
    - Columns 1716
    - Connect 943
    - Connector 851
    - Create From Attribute 896
    - Definition 1716
    - Element 1196
    - Elements And Connectors 709
    - Group, Toolbox 709
    - Instance 1196
    - Links 851
    - Move Between Packages 806
    - Multiplicity 964
    - Operations 1716
    - Packages 1716
    - Parameters 1716

- Object
  - Properties 843, 1716
  - Relationships 851
  - Scenario 852
  - State, Set 1199
  - Tables 1716
  - Transformation 1716
  - Type 1716
  - Type, Change For Element 809
- Object Constraint Language
  - Model Validation 2203
  - Model Validation Rules For Conformance 2209
- Object Diagram
  - Description 1015
  - Elements And Connectors 1015
  - Example 1016
- Object Flow
  - Connector 1250
  - In Activity Diagram 1250
  - In State Machine Diagram 1250
  - Multiple 1250
  - Relationship 1250
  - Selection Behavior 1250
  - Simple 1250
  - Transformation Behavior 1250
  - With Action Pins 1250
- Object Oriented Programming
  - C Code Generation For UML Model 1747
  - Limitations 1747
- Object Workbench 1861
  - Introduction 2176
  - Invoke Method 2179
  - Modes 2176
  - Overview 2176
  - Workbench Variables, Constructors 2177
  - Workbench Variables, Create 2177
- ObjectType Enum
  - Automation Interface 2419
- Occurrence
  - Connector 1252
  - Element 1139
  - Relationship 1252
- OCL
  - Model Validation 2203
- OCL Constraints
  - Attribute 2209
  - Element 2209
  - Feature 2209
  - Model Validation Rules for Conformance 2209
  - Relationship 2209
- ODBC
  - Data Modeling 1978
  - Driver, Set Up 200
  - ODBC Data Source
    - Select 1975
  - ODBC Source
    - Select Database Objects From 1976
    - Select Stored Procedures From 1976
    - Select Tables From 1976
  - ODM
    - Commands 1652
    - Example Diagrams 1651
    - Export OWL/RDF 1652
    - Import OWL/RDF 1652
    - MDG Technology For 1643
    - OMG 1643
    - OWL Definition Diagram 1651
    - OWL Elements, Descriptions 1646
    - OWL Facts Diagram 1651
    - OWL Properties, Descriptions 1646
    - OWL Relationships, Descriptions 1646
    - OWL Semantics, Descriptions 1646
    - RDF Definition Diagram 1651
    - RDF Elements, Descriptions 1649
    - RDF Relationships, Descriptions 1649
    - Toolbox Pages 1644
- Offline
  - Checkout 322
  - Version Control 322
- OLE DB Provider
  - Microsoft For Oracle 216
  - Oracle Provider 216
- on the File | Document Options menu 2261
- Online Resources
  - Access From Learning Center 89
  - Submenu 131
- Ontology Definition Metamodel
  - MDG Technology For 1643
- Open
  - External Tools 157
  - Package From Project Browser 680
  - Package Within Diagram 744
  - Report in Microsoft Word 2345
- Open Office
  - Optimize For, In RTF Report Generation 2256
- Open Project
  - Menu Option (File Menu) 92
- Open Repository
  - Automation Interface Code Example 2570
- Open Source Directory
  - Element Code 111
- Open Source File
  - Menu Option 115
- Operation

- Operation
- Add To Element, In-place Editor 924
  - Appearance 900
  - As Action 1091, 1100
  - Associate With Behavior 904
  - Behavior Description, Show In Diagram 904
  - Behavior, Initial Code 906
  - Business Rule 1534
  - Connect To 944
  - Constraints 906
  - Context Menu, Project Browser 582
  - Copy Between Elements 823
  - Create Use Case For 1807
  - Definition 900
  - Dialog, Behavior Tab 904
  - Dialog, General Tab 901
  - Dialog, Post Tab 906
  - Dialog, Pre Tab 906
  - Disconnect From 944
  - Display Inherited Operation 908
  - Edit Name, In-Place Editor 918
  - Edit Parameter Kind 923
  - Edit Scope 920
  - Edit Stereotype, In-place Editor 919
  - Element Feature 900
  - Fast Create 901
  - Field Substitution Macros 1431
  - Implement Interface Operations 908
  - Implement Parent Operations 908
  - In Project Browser 900
  - Inherited, Show 745
  - Introduction 900
  - Link To Use Case 1807
  - Move Between Elements 824
  - Of State Element, Behavior 904
  - Override Interface Operations 908
  - Override Parent Operations 908
  - Parameter Keyword, Edit 922
  - Parameter, By Reference 915
  - Parameter, Insert In Element 924
  - Parameters, Define 913
  - Private, Icon 585
  - Properties, Behavior 904
  - Properties, General 901
  - Properties, Initial Code 906
  - Properties, Postconditions 906
  - Properties, Preconditions 906
  - Protected, Icon 585
  - RuleTask, Behavior Call Action 1534
  - Show On Diagram 745
  - Tagged Values, Add 907
  - WSDL Port Type Operation 2076
  - Operation Constraints Dialog 2193
  - Operational Domain Model
    - SysML 2019
  - Options
    - Audit 384
    - Compare Utility 402
    - Element Visibility 559
    - Read-Only, For Enterprise Architect 26
    - Reset For A Class 1922
    - Visual Styles/Themes 162
  - Options Dialog
    - ActionScript 1909
    - Ada 2005 1909
    - Appearance Options 550
    - ArcGIS 1910
    - Attribute/Operation Specifications 1899
    - C 1911
    - C# 1912
    - C++ 1912
    - Communication Message Coloring 561
    - Connector Settings 559
    - Delphi 1913
    - Diagram Behavior 553
    - Diagram Settings 549
    - General Settings 544
    - Introduction 542
    - Java 1915
    - Links Settings 559
    - MDG Technology 1921
    - Object Appearance 556
    - Object Appearance, Default Fonts 552
    - Object Appearance, Element Visibility 559
    - PHP 1916
    - Python 1917
    - Sequence Diagram Options 555
    - Standard Colors 546
    - Standard Colors - Set Project Colors 548
    - SystemC 1917
    - VB.NET 1918
    - Verilog 1919
    - VHDL 1919
    - Visual Basic 1920
    - XML Specifications 562
  - Oracle
    - ODBC Driver, Set Up 208
    - Package, Create 1970
    - Sequence 1942
    - Sequence Options, DDL For Packages 1979
    - Sequence Options, DDL For Table 1978
    - Set Datatypes 1949
    - Tables, Set Properties 1940
    - Tables, Tagged Values 1940

## Oracle

- Temporary Table 1940
- Upsize To 230

## Oracle 10g

- Data Repository, Connect To, Via ODBC 214
- Data Repository, Connect To, Via OLE DB 216
- Server Repository, Create 197

## Oracle 11g

- Data Repository, Connect To, Via ODBC 214
- Data Repository, Connect To, Via OLE DB 216
- Server Repository, Create 197

## Oracle 9i

- Data Repository, Connect To, Via ODBC 214
- Data Repository, Connect To, Via OLE DB 216
- Server Repository, Create 197

## Order

- Enterprise Architect 35
- Package Contents 586

## Outline

- Red 276

## Output

- Business Modeling 1521
- Debugger, View 1875
- Debugger, Window 1875

## Output Window

- Audit History Tab 392
- Context Menu 164

## Override

- Attribute Initializer 895
- Default Toolbox In Toolbox Profile 1342
- Implement Parent Operations 908
- Parent Operations 908

## Overrides &amp; Implementations

- Of Classes And Interfaces 109

## Overview

- Visual Execution Analyzer 2143

## OWL

- Definition Diagram 1651
- Elements 1644
- Elements, Descriptions 1646
- Example Diagrams 1651
- Export To File 1652
- Facts Diagram 1651
- Import From File 1652
- Profile 1643
- Properties 1644
- Properties, Descriptions 1646
- Relationships 1644
- Relationships, Descriptions 1646
- Semantics 1644
- Semantics, Descriptions 1646
- Toolbox Pages 1644

**- P -**

## Package

- Add And Manage, Automation Interface Code Example 2571
- Add Element Directly 798
- Add To Diagram From Toolbox 681
- Add To Project Browser 681
- Add To UML Model, Quick Start 52
- Apply Version Control To Model Branch 371
- As RTF Bookmark 2346
- Automation Interface 2390
- Automation Interface Repository 2433
- Baselines 394
- Batch Export To XMI 430
- Batch Import From XMI 431
- Body, For Oracle 1970
- Comparison, Example 406
- Configuration 426
- Configure For Version Control 362
- Context Menu, Project Browser 571
- Control, Menu 424
- Control, Remove 428
- Controlled 362
- Controlled, Report Deletion Of Cross Package References in XMI 433
- Copy 683
- Copy Between Projects 444
- Create Diagram 724
- Create Oracle Packages 1970
- CSV Import/Export Specification 435
- Delete In Project Browser 685
- Duplicate 683
- Element 1200
- Existing, Drag Onto Diagram 684
- Export 422
- Export To XMI Stubs 426
- Export Version Controlled Model Branch 371
- Field Substitution Macros 1432
- Hide Contents On Diagram 685
- Icon Overlays 585
- Import 422
- Import Version Controlled Model Branch 372
- Load 429
- Lock 273
- Lock, Require User Lock 274
- Lock, User/Group Lock 271
- META-INF 1694
- Modeling With 680
- Move 57
- Move Element Between 806



- Package
  - Move Objects Between 806
  - Nested in Version Control 320
  - Open From Diagram 744
  - Open From Project Browser 680
  - Paste Copy 683
  - Phase, Update 529
  - Profile 1296
  - Publish 414
  - Rename 682
  - Resynchronize Package Version Control Status 380
  - Review Version Control History (SCC) 377
  - Review Version Control History (TFS, CVS, Subversion) 376
  - Save 428
  - Save Profile 1307
  - Show Contents On Diagram 685
  - Specification, For Oracle 1970
  - Status, Update 529
  - Synchronize Contents 1774
  - Tasks 680
  - Update Contents 1774
  - Validate Version Control Configuration 379
  - Version Control 312
  - Version Control Revision, Check Out 376
  - Version, Update 529
  - Working With 680
  - XML 422
- Package Browser
  - Context Menu Options 592
  - Description 587
  - Filter Bar 590
  - Generate RTF Report 2253
  - Insert Tagged Value Column 592
  - Options 592
  - Toolbar 592
  - Value Grouping 590
  - View Header 590
  - Work On Elements 592
- Package Context Menu
  - Add Submenu, Project Browser 574
  - Build And Run Submenu, Project Browser 576
  - Code Engineering Submenu, Project Browser 575
  - Contents Submenu, Project Browser 577
  - Documentation Submenu, Project Browser 575
  - Import/Export Submenu, Project Browser 577
- Package Control
  - Options, Dialog 426
- Package Diagram
  - Description 1011
  - Elements And Connectors 1011
  - Example 1013
- Package Import
  - Connector 1252
  - Relationship 1252
- Package Merge
  - Connector 1253
  - Relationship 1253
- Package Scenarios
  - Import As Test Scenarios 2224
- Package Structure
  - BPEL 1.1 1573
  - BPEL 2.0 1597
- Packaging Component
  - Element 1674
- Page
  - Footer, Print On Diagram 727
  - Header, Print On Diagram 727
  - Setup, Menu Option (File Menu) 92
  - Size, Set For Diagram 764
- Pan
  - Diagram View 763
- Pan And Zoom
  - Diagram 610
  - Window 610
- Paragraph Numbering
  - In RTF Style Template Editor 2285
- Parameter 915
  - Activity 913
  - Automation Interface, ElementFeatures Package 2406
  - Behavior 913
  - Behavior Call Arguments 912
  - Behavior, Tagged Values 914
  - Behavioral, Edit 912
  - Behavioral, Extend 912
  - Behavioral, Reassign 912
  - Behavioral, Set 912
  - Behavioral, Synchronize Call Argument With 911
  - Behavioral, Synchronize Invocation Argument With 911
  - Dialog 913
  - Field Substitution Macros 1433
  - For Rule Flow Activity 1533
  - Interaction 913
  - Kind Inout, By Reference 915
  - Operation 913
  - Operation, By Reference 915
  - Reference 915
  - Show Details On Diagram 732
  - Substitution 1258

- Parameter 915
  - Variables In Business Rules 1533
- Parameter Information
  - Display In Source Code 1802
- Parameter Kind
  - Edit for Element Operation 923
- Parameter Substitution
  - Binding Connector 1258
- Parameterized Class
  - And Binding Connector 1257
- Parameterized Classes (Templates) 1183
- Parametric
  - Diagram 2013
  - Diagram, Simulate SysML Model 2016
  - Model, SysML, Create 2013
- ParamTag
  - Automation Interface, ElementFeatures Package 2408
- Parent
  - Confirm Element As Parent 805
  - Set For Element 801
- Parent And Interfaces
  - Elements 109
- Parse
  - Source Code Files In Viewer 1799
- Part
  - Add Property Value 1201
  - Element 1200
  - Property Tab 1204
  - Qualifiers 1204, 1212
  - Redefined 1204
  - Represent On Sequence Diagram 1084
  - Subsetted 1204
- Partial Class
  - Generate 1772
- Partition
  - Activity 1148
  - Docking 1148
  - Element 1148
  - Horizontal 1148
  - Vertical 1148
- Partitions Collection
  - Automation Interface, ElementFeatures Package 2409
- Pascal Case
  - Naming Format 1728
- Password
  - Administrator Change 270
  - Administrator Set 270
  - Security, Change 256
  - User Change 270
- Password Encryption
  - Prior To Release 7.1 Of Enterprise Architect 269
- Password Encryption (Repository)
  - At Release 7.1 Of Enterprise Architect 183, 188
- Paste
  - Activity As Action 736
  - Activity As Link 736
  - Composite Element 736
  - Connectors Between Pasted Activities 736
  - Connectors Between Pasted Instances 733
  - Copy Of Element 808
  - Copy Of Package 683
  - Diagram Into Package 742
  - Element From Project Browser 733
  - Element On Diagram 737
  - Multiple Items As Children 735
  - Multiple Items As Instances 735
  - Multiple Items As Links 735
  - Multiple Items From Project Browser 735
  - Object As Child 733
  - Object As Link 733
  - Object As New Instance 733
- Paste Element Dialog 733
- Paste Elements
  - As Link 94
  - As New 94
  - From Clipboard As Metafile 94
  - Menu Option (Edit Menu) 93
  - Submenu (Edit Menu) 94
- Patches
  - SQL, Run 539
- Pattern
  - Action, Modify 1274
  - Actions 1271
  - Add To MDG Technology 1328
  - Application, Generate 1738
  - Create From Diagram 1271
  - Default, Change 1274
  - Design 1271
  - Gang of Four 1271
  - GoF 1271
  - GoF, Download 1273
  - Import Into Model 1273
  - In Resources View 1271
  - Model-View-Controller 1665
  - MVC 1665
  - Save 1271
  - Save From Diagram 1271
- PDATA
  - Diagram Profile Attribute Values 1348

- PDATA  
   Element Attribute In MiscData, Object Model 2478
- PDF  
   Output Report In 2256
- People  
   As Project Resources 992  
   Assign To Changes 2247  
   Assign To Defects 2247  
   Dialog 989  
   Project Types Menu Option 129  
   Settings 989
- Per Page Diagram Layout 776
- Performance  
   Of Enterprise Architect 210, 212, 214, 216, 218, 219, 221, 238, 382
- Permission List  
   User Security 264
- Personal Information  
   Allocated Work 486  
   Mail, Sent & Received 486  
   Project Tasks 486  
   Window 486  
   Workflow 486  
   Working Sets 486
- Personal Information Window  
   Allocated Work Tab 493  
   Execute Workflow Scripts 498  
   Monitor Tasks 496  
   Project Tasks Tab 496  
   Record Work Progress 493  
   Resource Allocation 493  
   Workflow Tab 498  
   Working Sets Tab 498
- PHP  
   Code Generation 1916  
   Import, Reverse Engineering 1793  
   Language Options 1916  
   Modeling Conventions 1758
- PHP Apache  
   Sample Model, Visual Execution Analysis 2145
- PHP Debugger  
   Checklist 1857  
   Local 1854  
   Remote 1854  
   Scenarios 1854  
   Script Set Up 1854  
   Supported Operating Systems 1857  
   System Requirements 1857  
   Timeouts 1857
- Physical Data Model 1932
- Physical Model 662, 671
- PI  
   Macro 1448  
   Processing Instruction 1448
- PI Macro  
   Code Template Syntax 1444
- PIM  
   Internal Bindings 1679
- Pin  
   Action 1097  
   Add To Action 1097  
   As Action Property 1098  
   As Argument For Call Action 1097  
   Assign To Action 1097, 1098  
   Connector End Point 148  
   Connector Start Point 148  
   Properties 1097  
   Toolbox Pages 701
- Pkg Import  
   Connection 1252  
   Relationship 1252
- Pkg Merge  
   Connector 1253  
   Relationship 1253
- Place Recording Markers  
   Execution Analysis, Recording Sequence Diagrams 2154
- Platform Naming Conventions 1728
- Platform Specific Model 1679
- Platform-Independent Model 1679
- Pool  
   BPMN, Use By BPEL 2.0 Model 1612
- Pool (BPEL 1.1)  
   Create 1588  
   Model 1588
- Port  
   Add To Element 1203  
   Element 1202  
   For Trigger Element 1173  
   Inherited By Component Instance, From Classifying Component 1203  
   Inherited By Component, From Realized Interface 1203  
   Property Tab 1204  
   Qualifiers 1204, 1212  
   Redefined 1203, 1204  
   Represent On Sequence Diagram 1084  
   Subsetted 1204
- Port Type  
   WSDL Diagram 2075  
   WSDL Element 2075
- Port Type Operation  
   WSDL 2076

- Position
  - Options For Elements In Diagram 108
- Position Elements 112
- Post
  - Add To Team Review 287
  - Create In Team Review 287
  - Create Reply 289
  - Delete 280
  - Delete In Team Review 291
  - Edit In Team Review 291
  - Reply To In Team Review 289
- PostgreSQL
  - Data Repository, Connect To 218
  - ODBC Driver, Set Up 202
  - Repository, Create 197
  - Upsize To 232
- Post-New Events
  - Add-In Model 2634
  - EA\_OnPostNewAttribute 2639
  - EA\_OnPostNewConnector 2636
  - EA\_OnPostNewDiagram 2637
  - EA\_OnPostNewDiagramObject 2638
  - EA\_OnPostNewElement 2635
  - EA\_OnPostNewGlossaryTerm 2642
  - EA\_OnPostNewMethod 2640
  - EA\_OnPostNewPackage 2641
- Pre/Post Conditions
  - Constraints 1093
  - Local 1093
  - Notes 1093
  - On Actions 1093
- Precede
  - Connector 707
- Predefined Tag Type
  - Assign To Stereotype 1300
  - Define 1300
- Predefined Tagged Value Type
  - Filters 1386
  - Reference Data 1390
  - Structured 1386
  - Syntax 1386, 1390
- Pre-Deletion Events
  - Add-In Model 2642
  - EA\_OnPreDeleteAttribute 2644
  - EA\_OnPreDeleteConnector 2646
  - EA\_OnPreDeleteDiagram 2647
  - EA\_OnPreDeleteDiagramObject 2647
  - EA\_OnPreDeleteElement 2643
  - EA\_OnPreDeleteGlossaryTerm 2649
  - EA\_OnPreDeleteMethod 2645
  - EA\_OnPreDeletePackage 2648
- Pre-New Events
  - Add-In Model 2650
  - EA\_OnPreNewAttribute 2655
  - EA\_OnPreNewConnector 2652
  - EA\_OnPreNewDiagram 2653
  - EA\_OnPreNewDiagramObject 2654
  - EA\_OnPreNewElement 2651
  - EA\_OnPreNewMethod 2656
  - EA\_OnPreNewPackage 2657, 2658
- Preprocessor Macros 1902
- Preserve Hierarchy
  - CSV Specification 438
- Primary Key
  - Complex 1952
  - Create 1952, 1953
  - Description 1952
  - Extended Properties 1955
  - Name Template, Define 1952, 1954
  - Simple 1952
  - SQL Server, Non-Clustered 1955
- Primitive
  - Element 1204
- Print
  - Diagram (Single) From Project Browser 581
  - Diagrams (Multiple) From Project Browser 581
  - Menu Option (File Menu) 92
  - Multi-page Diagrams 766
  - Page Footer On Diagram 727
  - Page Header On Diagram 727
  - Preview, Menu Option (File Menu) 92
  - Project Issues 462
  - Scaled Image 766
  - Setup, Menu Option (File Menu) 92
  - Task List 460
- Print Preview
  - Display 690
  - Multiple Pages 690
- Private Key
  - Add 2739
- Problem
  - Field Substitution Macros 1434
- Problem Type
  - Define 1001
- Process
  - BPEL 1.1, Properties 1576
  - BPEL 2.0, Model 1599
  - Element 1518, 1674
  - Element (Data Flow Diagram) 1506
  - Model Template 662, 665
  - Modeling 662, 665
- Process First Chance Exceptions 1889
- Process Memory
  - Inspect 1886

- Process Modeling Notation 1518
- Processing Instruction
  - PI Macro 1448
- Professional Edition
  - Of Enterprise Architect 22
  - Upsize From 193
- Profile
  - Add Attribute To Diagram 1280
  - Add Connector To Diagram 1280
  - Add Element To Diagram 1280
  - Add Elements 1297
  - Add Enumeration Elements 1303
  - Add Metaclasses 1297
  - Add Operation To Diagram 1280
  - Add Shape Script 1304
  - Add Stereotypes 1297
  - Add To MDG Technology 1327
  - And Element Templates 820, 1277, 1293
  - Application, Analyzer menu Option 120
  - Constraints 1302
  - Create 1295, 1296
  - Diagram, Create 1346
  - Elements And Connectors 714
  - Export To Disk 1306
  - Group, Toolbox 714
  - Import 1279
  - Import From XML 1277, 1293
  - Learning Center, Create 1350
  - Learning Center, Link To Document Files 1351
  - Learning Center, Run Add In Functionss 1353
  - Package 1296
  - Relationship Matrix, Render In RTF Report 2310
  - Save 1306
  - Save From Diagram 1307
  - Save From Package 1307
  - Set Default Appearance Of Stereotype Objects 1306
  - Stereotype 1296, 1316
  - Stereotypes 1277, 1293
  - Swimlanes Matrix 753
  - Tagged Values 1280
  - Tags 1299
  - Toolbox 1337
  - Toolbox, Add Custom Compartments 1339
  - Work With 1295
  - Zachman 753
- Profile Connector
  - Add To Diagram 1280
- Profile Diagram
  - Description 1025
  - Example 1026
- Profile Element
  - Add To Diagram 1280
- Profiler
  - Attach To Process 2173
  - Getting Started 2171
  - Launch 2173
  - Operation 2174
  - Overview 2170
  - Prerequisites 2171
  - Report, Example 2170
  - Report, Load 2175
  - Report, Save 2175
  - Report, Save As Resource In Team Review 2175
  - Set Options 2174
  - Set Sample Intervals 2174
  - Start 2173
  - Stop 2173
  - Supported Platforms 2171
  - System Requirements 2171
  - Team Review, Save Report As Resource 2175
  - Toolbar 2171
- Progress OpenEdge
  - Data Repository, Connect To 221
  - ODBC Driver, Set Up 206
  - Server Repository, Create 199
  - Upsize To 236
- Project
  - Administration, Security Permissions 253
  - Author 989
  - Browser 567
  - Change Control 2233
  - Clean 535
  - Clean Prior To Build 1833
  - Clients 993
  - Compact .EAP File 539
  - Comparison With Other Project 445
  - Configure 189
  - Constants 2268
  - Copy Existing 191
  - Create In Enterprise Architect, Tutorial 49
  - Create, Quickstart Tutorial 49
  - Data Integrity 535
  - Data, Transfer 442
  - Develop In Team Environment, Introduction 240
  - Development, Introduction 176
  - EABase 47
  - Estimation 516
  - Explorer 567
  - File, EABase 189
  - Glossary 460, 467
  - Integrity Check 535

## Project

- Integrity, Run SQL Patches 539
- Issues 460
- Issues, Record 462
- Items, Add Via Toolbar 141
- Maintenance 2233
- Menu 100
- Metrics 516
- Model Template 674
- Open Existing 178
- Recover 535
- Remove Recent 89
- Rename .EAP File 539
- Repair Project .EAP File 540
- Resources 992
- Roles 991
- Share, DBMS Repository 242
- Share, Network Drive 242, 243
- Share, Replication 242
- Sharing, Introduction 240
- Spell Checking 480
- Statistics, View 100
- Structure 657
- Tasks 72, 460
- Timescale Estimation 516
- Toolbar 141
- Transfer 224
- Upgrade 537
- View 567
- What Is A? 179
- Project Browser 577
  - Attribute Context Menu 582
  - Collapse Contents 577
  - Context Menus 569
  - Default Behaviour 587
  - Diagram Context Menu 581
  - Element Context Menu 577
  - Element Context Menu, Add Submenu 580
  - Exclamation Marks 276
  - Expand Contents 577
  - Free Sorting 587
  - Hide And Show 567
  - Icon Overlays 585
  - Icon, User-Defined 1308
  - Introduction 567
  - Method Context Menu 582
  - Model Context Menu 569
  - Move Items Within 567
  - Open Package 680
  - Operation Context Menu 582
  - Order Package Contents 567
  - Package Context Menu 571
  - Package Context Menu, Add Submenu 574
  - Package Context, Build And Run Submenu 576
  - Package Context, Code Engineering Submenu 575
  - Package Context, Contents Submenu 577
  - Package Context, Documentation Submenu 575
  - Package Context, Import/Export Submenu 577
  - Reload Current Package 577
  - Reset Sort Order 577
  - Scripts 2375
  - Selective Collapse of Packages 567
  - Show Level Numbering 571
  - Show Stereotypes 587
  - Toolbar 583
  - Version Control Indicators 321
  - Views 567
- Project Calendar
  - Allocated Resources View 510
  - Calendar Panel 502
  - Calendar Panel, Configure Event Subtypes 509
  - Diary Panel 502
  - Events 505
  - Options 502
  - Overview 502
  - Project Tasks View 513
  - Recurring Events 505
  - Toolbar 502
  - View, Calendar 505
- Project Constant
  - RTF Reports 2268
- Project Custom Colors
  - Get For Element 819
  - Set For Element 819
- Project Discussion Forum
  - Now Team Review 279
- Project Factor Calibration 516
- Project File
  - Create 178
  - Example 181
  - Open 181
- Project Glossary
  - Add Item, Glossary Detail Dialog 468
  - Add Item, Glossary Dialog 469
  - Create Item From Notes Text 977
  - Delete Item, Glossary Dialog 469
  - Delete Item, Project Glossary Tab 468
  - Filter List, Glossary Dialog 469
  - Filter List, Project Glossary Tab 468
  - Glossary Dialog 469
  - Glossary Report 471

- Project Glossary
  - Hyperlink Term From Notes 977
  - Insert Item In Text 977
  - Modify Item, Glossary Detail Dialog 468
  - Modify Item, Glossary Dialog 469
  - Redefine Entry Type, Project Glossary Tab 468
- Project Indicators
  - Project Types Menu Option 129
  - Risk Types 458
- Project Interface
  - Automation Interface 2534
  - Project 2535
- Project Issue
  - Add 463
  - Delete 463
  - Dialog 462
  - Modify 463
  - Print 462
  - Record 462
  - Report, Via Project Issues Dialog 465
  - Report, Via Project Issues Tab 466
  - Tab 462
- Project Management
  - Asterisk On Window Tabs 449
  - Default Hours 520
  - Effort Management 452
  - Effort Types 456
  - Environment Complexity Factors 518
  - Introduction 447
  - Maintenance 2236
  - Metric Types 457
  - Metrics 448, 455
  - Model Template 662
  - Resource Allocation 450
  - Resource Report 477
  - Resources 448
  - Risk Management 453
  - Risk Types 458
  - Risks 448
  - Technical Complexity Factors 517
  - Toolbar 448
  - Window 448, 449
  - With Enterprise Architect 447
- Project Manager
  - And Enterprise Architect 80
  - Project Estimation 80
  - Project Role 80
  - Resource Management 80
  - Risk Management 80
- Project Role
  - And Enterprise Architect 71
- Business Analyst 74
- Database Administrator 84
- Deployment and Rollout 81
- Developer 78
- Implementation Manager 81
- Project Manager 80
- Software Architect 75
- Software Engineer 77
- Technology Developer 82
- Tester 80
- Project Settings
  - Configure (Settings Menu) 127
- Project Task
  - Add 461
  - Delete 461
  - Description 460
  - List 460
  - Modify 461
  - Tab, Print List 460
- Project Task Allocation
  - Element View 475
  - Resource View 473
  - Resource Report 477
  - Window 472
- Project Tasks
  - Create, Project Calendar 513
  - Create, Personal Information Window 496
  - Delete, Project Calendar 513
  - Delete, Personal Information Window 496
  - Edit, Project Calendar 513
  - Edit, Personal Information Window 496
  - Monitor, Project Calendar 513
  - Monitor, Personal Information Window 496
  - Tab, Personal Information Window 496
  - View, Project Calendar 513
- Project Team Review
  - Access 279
  - Add External File Link To Post 292
  - Add New Category 284
  - Add New Post 287
  - Add Object Link To Post 292
  - Add Team Review Link To Post 292
  - Connections To Other Team Reviews 296
  - Copy Path To Clipboard 280
  - Create Category 284
  - Create Post 287
  - Create Reply 289
  - Delete Item 291
  - Edit Item 291
  - Editor 291
  - Icons 279
  - Introduction 279

- Project Team Review
  - Load Data When Required 296
  - Loading Behavior 296
  - Mark All Posts Unread 296
  - Options 296
  - Preload 296
  - Reply To Post 289
  - Search 295
  - Was Discussion Forum 279
- Project Types
  - Estimation Factors 129
  - General Types 129
  - Maintenance Data 129
  - People 129
  - Project Indicators 129
  - Submenu 129
- Project Version Control Menu 355
- ProjectIssues
  - Automation Interface Repository 2442
- ProjectResource
  - Automation Interface Repository 2444
- ProjectRole
  - Automation Interface Repository 2445
- Properties
  - Automation Interface, ElementFeatures Package 2410
  - Behavior Tab 910
  - Call Tab 910
  - Connector, Advanced 968
  - Connector, Menu Section 937
  - Constraints, Scenario 875
  - Dialog, Element 843
  - Editor Language 1896
  - Effect Tab 1091
  - Element 843
  - Element Context Menu 828
  - Element, Advanced 847
  - Element, Associated Files 875
  - Element, Connectors 851
  - Element, Constraints 850
  - Element, Details 846
  - Element, External Requirements 850
  - Element, General 845
  - Element, Internal Requirements 848
  - Element, Links 851
  - Element, Relationships 851
  - Element, Requirements 848
  - Element, Trigger Tab 1173
  - Extend For Requirements 1478
  - Object 843
  - Of Classifiers, Composite Structure Diagram 1019
  - Of Requirements, Extended 1478
  - Of Requirements, Standard 1475
  - Requirement, Display On Diagram 1478, 1479
  - Window (Element) 878
- Properties Note
  - Diagram 747
- Properties,
  - Part, Property Tab 1204
  - Port, Property Tab 1204
- Properties, EASL
  - Action 1454
  - Argument 1454
  - Behavior 1454
  - Call Event 1454
  - ChangeEvent 1454
  - Classifier 1454
  - Condition 1454
  - Construct 1454
  - Edge 1454
  - EventObject 1454
  - Instance 1454
  - Parameter 1454
  - Primitive 1454
  - PropertyObject 1454
  - SignalEvent 1454
  - State 1454
  - StateMachine 1454
  - TimeEvent 1454
  - Transition 1454
  - Trigger 1454
  - Vertex 1454
- Property
  - Automation Interface, ElementFeatures Package 2410
  - Create In BPEL 2.0 1611
  - Model In BPEL 2.0 1611
  - Redefined 882
  - Select, Dialog 882
  - Subsetted 882
- Property Validation
  - Attribute 2208
  - Element 2208
  - Feature 2208
  - Relationship 2208
- Property Value
  - Part, Add To 1201
- PropertyType
  - Automation Interface Repository 2446
- PropType Enum
  - Automation Interface 2421
- Proxy
  - (Shortcut) File 183



- Pseudo-State
    - Elements 1037
    - Fork 1130, 1131
    - In State Machine Diagram 1037
    - Join 1130, 1133
  - PSM 1679
  - Publish Model Package
    - For ArcGIS XML Workspace 1640
    - For XMI 414
  - Purchase
    - Enterprise Architect 35
  - Python
    - Code Generation 1917
    - Import, Reverse Engineering 1793
    - Language Options 1917
    - MDG Technology For, Enterprise Architect 1289
    - Modeling Conventions 1759
- Q -**
- Qualified Association 1211
  - Qualifier
    - Association End 1212
    - Association Property 1211
    - Attribute 1212
    - Attribute Property 1211
    - Dialog 1212
    - Of Attribute 889
    - Of Port 1204
    - Part 1212
    - Part Property 1211
    - Port 1212
    - Port Property 1211
    - Set Properties 1212
  - Quality Control 2202
    - Model Validation 2203
    - Spell Checking 480
    - Testing 2211
  - Query Builder
    - Search Definition 623
  - Query Methods
    - In Shape Scripts 1373
  - Quick Add
    - Tagged Values 971
  - Quick Linker 943
    - Arrow 791
    - Connector Names, In Definitions 1322
    - Create Connector 793
    - Create Element And Connector 792
    - Default Settings, Hide 1322
    - Definition Format 1318
    - Element Names, In Definitions 1322
    - Example 1320
    - Introduction 791, 1317
    - Options 792
  - Quick Start
    - Add Connectors To UML Model 55
    - Add Diagram To Package 52
    - Add Diagram To UML Model 52
    - Add Element To Diagram 53
    - Add Element To UML Model 53
    - Add Package to UML Model 52
    - Auditing 383
    - Create A Project In Enterprise Architect 49
    - Define Connector Properties 56
    - Define Element Properties 56
    - Delete UML Model Components 62
    - Generate HTML Report 2353
    - Generate RTF Report 2253
    - Move Project Components 57
    - Project Tasks 72
    - Save Project Changes 67
    - View HTML Report 2353
  - Quickstart
    - Common Tasks 45
- R -**
- Rational Rose
    - And XMI 411
    - Export To 421
  - Rational Software Architect
    - Models, Import 419
  - Rational Software Modeler
    - Import \*.emx Files 417
    - Import \*.uml2 Files 417
  - RDF
    - Definition Diagram (ODM) 1651
    - Elements 1644
    - Elements, Descriptions 1649
    - Example Diagram 1651
    - Export To File 1652
    - Import From File 1652
    - Profile 1643
    - Relationships 1644
    - Relationships, Descriptions 1649
    - Toolbox Pages 1644
  - Read-Only Options
    - For Enterprise Architect 26
  - Realization
    - Connector, Quick Generation Of 1480
  - Realization Link

- Realization Link
  - Implement Parent Operations 908
  - Override Parent Operations 908
- Realize
  - An Information Flow 1229
  - Connector 1254
  - Relationship 1254
- Realized Interfaces
  - For Class, Show On Diagram 760
- Receive
  - Element 1149
  - Event 1667
- Recent Discussions
  - Concerning Model View Items 598
- Recent Post Options Dialog 601
- Recent Postings
  - Display 604
- Reception
  - Definition 1205
  - Of Signal 1205
- Record
  - Application, Analyzer menu Option 120
  - Submenu (Analyzer) 123
- Record & Analyze
  - Window Context Menu 2151
  - Window, Toolbar 2163
- Record & Analyze Window 2149
- Record Activity For Class
  - Execution Analysis, Record Sequence Diagram 2160
- Record Activity For Method
  - Execution Analysis, Record Sequence Diagram 2155
- Record Macro
  - In Source Code Editor 1802
- Record Marker
  - Set 2155
- Record Sequence Diagrams
  - Control Recording 2161
- Record State Changes
  - Visual Execution Analyzer 2168
- Record Work Progress
  - Personal Information Window 493
- Recorder
  - Toolbar 2163
- Recording Actions
  - Create Sequence Diagram, Call Stack 1886
  - Debugger, Step Through Function Calls 2162
- Recording History
  - Context Menu Options 2151
  - Save, Execution Analysis 2164
  - Work On 2151
- Recording Markers
  - Activate, Execution Analysis 1869
  - Breakpoints And Markers Window, Execution Analysis 1869, 2157
  - Difference From Breakpoint 2155
  - Disable, Execution Analysis 1869
  - Marker Types, Execution Analysis 2155
  - Place, Execution Analysis 2154
  - Work With Marker Sets, Execution Analysis 2158
- Recover
  - Controlled Package 432
  - Diagram After Changes (Baseline) Project 403
  - Project 535
- Rectangle Notation
  - Element Menu Option 829
  - For Shaped Elements 1177
- Recursion
  - Connector 1255
  - Message 1255
  - Relationship 1255
- Recursive Builds
  - Visual Execution Analyzer 1832
- Red
  - Border 276
  - Exclamation Mark 276
  - Letter A 928, 929
  - Object Outline 276, 767, 837
  - Triangle 530
- Redefined
  - Property, Port 1204
- Redefined Port
  - Manage 1203
- Redefined Property
  - Of Attribute 889
- Redefinition
  - Connector 714
- Redo
  - Last Action, Diagram Edits 768
  - Option (Edit Menu) 93
- Re-entrancy
  - In Add-Ins 2591
- Reference
  - Automation Interface 2389
  - Automation Interface Repository 2447
- Reference Data 981
  - And Version Control 317
  - Cardinality (Multiplicity) 986
  - Clients 993
  - Constraint Status Types 998
  - Constraint Types 997
  - Estimation 1000

- Reference Data 981
  - Export 306
  - Export, Introduction 305
  - General Types 995
  - Import Automatically 308
  - Import Manually 308
  - Import, Introduction 305
  - Maintenance 1001
  - Metrics 1000
  - People 989
  - Problem Types 1001
  - Project Author 989
  - Requirement Types 998
  - Resources 992
  - Roles 991
  - Scenario Types 999
  - Share 305
  - Status Types 995
  - Stereotypes 982
  - Tagged Value Types 985
  - Testing Types 1002
  - UML Types 981
- Reference Data Tagged Value Type 1390
- Reference Data Tagged Values
  - Create 1392
- Referenced XML Schema
  - Import 2053
- Refresh
  - Diagram 242
  - Image 755
  - Project 242
  - View Of Shared Model 242
- Region
  - Composite State 1037
  - Concurrent Substate 1037
  - Element 1150
  - Expansion, Element 1124
  - Interruptible Activity, Element 1141
  - On Composite State 1153
  - State Machine 1037
- Register
  - Add-Ins 2745
  - Enterprise Architect 38
- Registration Key
  - In License Information 2738
- Registry Settings 153
  - User 542
- Regular Expressions
  - Code Editor Find And Replace 1814
  - Metasequences 1814
  - Tagged Regions 1814
- Related Elements
  - Find 738
  - Insert With Context Menu 833
  - Place On Current Diagram 738
- Relational Database
  - Model Template 662, 669
- Relationship 1627
  - Activity Edge 1230
  - Aggregate 1208
  - Application 714
  - ArcGIS 1634
  - ArchiMate 1621
  - Assembly 1209
  - Associate 1210
  - Association 1210
  - Association Class 1215
  - Asynchronous Signal 1241
  - Communication 1218
  - Communication Path 1217
  - Compose 1217
  - Composite Aggregation 1217
  - Connector 1218
  - Control Flow 1219
  - Create Using Relationship Matrix 644
  - Data Flow 1506
  - Delegate 1220
  - Delete 851
  - Delete Using Relationship Matrix 644
  - Dependency 1221
  - Dependency, Apply Stereotype 1221
  - Deployment 1222
  - Display Options 559
  - Element (Entity Relationship Diagram) 1627
  - Entity Relationship Diagram 1627
  - Eriksson-Penker 1623
  - Extend 1223
  - Extension (Profile Toolbox) 714
  - Generalization 1224
  - Generalize 1224
  - Generalize (Profile Toolbox) 714
  - Hide 851
  - Implements 1254
  - Include 1224
  - Information Flow 1225, 1226
  - Inheritance 1224
  - Interrupt Flow 1230
  - List, On Context References Tab 874
  - Manifest 1231
  - Matrix, In Traceability 636
  - Message 1231
  - Mind Mapping 1503
  - Modify Using Relationship Matrix 644
  - Nesting 1249

- Relationship 1627
  - Notelink 1249
  - Object Flow 1250
  - Occurrence 1252
  - Package Import 1252
  - Package Merge 1253
  - Pkg Import 1252
  - Pkg Merge 1253
  - Realize 1254
  - Recursion 1255
  - Redefinition 714
  - Representation 1257
  - Represents 1256
  - Role Binding 1255
  - Self Message 1235
  - Show 851
  - SysML Activity 1992
  - SysML Block Definition 1996
  - SysML Interaction 1999
  - SysML Internal Block 2000
  - SysML Model 2002
  - SysML Parametric 2005
  - SysML Requirement 2007
  - SysML State Machine 2009
  - SysML Use Case 2011
  - Tagged Value (Profile Toolbox) 714
  - Trace 1260
  - Transition 1260
  - Usage 1263
  - Use 1263
  - Visibility 953
  - Window, Context Menu 648
- Relationship Matrix
  - Access 640
  - Access From Shortcut 183, 187
  - Access From Shortcut (Direct Definition) 185
  - Create Relationship 644
  - Delete Relationship 644
  - Export to CSV 645
  - Incorporate In HTML Report 645
  - Incorporate In RTF Report 645
  - Introduction 639
  - Link Direction 641
  - Link Type 641
  - Locate Elements 647
  - Manage Display Content 645
  - Modify Relationship 644
  - Open 640
  - Options 645
  - Print Matrix 645
  - Print Preview 645
  - Profile, Delete 643
  - Profile, Render In RTF Report 2310
  - Profile, Save 643
  - Profile, Update 643
  - Review Elements 647
  - Save As .EMF File 645
  - Save As .PNG File 645
  - Scale Printout Width 645
  - Set Element Type 640
  - Set Link Direction 641
  - Set Link Type 641
  - Set Source Package 641
  - Set Target Package 641
- Relationships Window
  - Context Menu 738
- Reload
  - Diagram 242
  - Model (Shared) 242
  - Project 242
  - View 242
- Reload Current Project
  - Menu Option (File Menu) 92
- ReloadType Enumeration
  - Automation Interface 2421
- Remove
  - Package Control 428
  - Recent Project 89
  - Replication 248
- REMOVE\_PREFIX
  - Macro 1728
- Rename
  - Diagram 739
  - Package 682
  - Project .EAP File 539
  - Views 678
- Re-Order
  - Messages 1244
- Reorder Columns 1944
- Repair
  - Project .EAP File 540
- Replace
  - Linked Document 933
- Replica
  - Create 246
  - Synchronize 247
  - Upgrade 249, 538
- Replication
  - And Version Control 353
  - Change Collisions 244
  - Create Design Masters 246
  - Create Replicas 246
  - Disable 244
  - Impact On Auto Increment Columns 536

- Replication
  - Introduction 244
  - Merge Rules 244
  - Remove 248
  - Resolve Conflicts 250
  - Synchronize Replicas 247
  - Upgrade Replicas 249
  - Using 244
- Reply
  - To Post In Team Review 289
- Report
  - Dependency 2339
  - HTML 2249, 2352
  - Implementation 2341
  - Open In Microsoft Word 2345
  - Project Issues, Via Project Issues Dialog 465
  - Project Issues, Via Project Issues Tab 466
  - Rich Text Format 2249, 2251
  - RTF 2249, 2251
  - Testing Details 2343
- Reporting
  - Dependency 1492
  - Implementation 1492
  - Requirements 1492
- Repository
  - Access Permissions For 193
  - Adaptive Server Anywhere, Create 198
  - Attributes 2448
  - Author Collection 2425
  - Automation Interface 2448
  - Client Collection 2425
  - Collection Class 2427
  - Connect To 193
  - Create 193
  - Datatype 2429
  - Encrypt Password 183, 188
  - EventProperties 2431
  - EventProperty 2432
  - Methods 2448
  - ModelWatcher 2432
  - MSDE Server, Create 198
  - Open, Automation Interface Code Example 2570
  - Package 2433
  - Package, Automation Interface 2423
  - Progress OpenEdge, Create 199
  - ProjectIssues 2442
  - ProjectResource 2444
  - PropertyRole 2445
  - PropertyType 2446
  - Reference 2447
  - Set Up Database 193
  - Stereotype 2470
  - Task 2471
  - Term 2473
  - Transfer Data Between 442
  - Use Extras, Automation Interface Code Example 2578
- Representation
  - Connector 1257
  - Relationship 1257
- Represents
  - Connector 1256
  - Relationship 1256
- Require User Lock
  - Apply User Lock 274
  - Release User Lock 274
- Require User Lock Policy 255
- Requirement
  - Aggregation 1480
  - Analysis 662, 666
  - And Level Numbering 1483
  - And Use Cases 1483
  - Auditing 1491
  - Automation Interface, Element Package 2495
  - Autonumbering 1473
  - Baselines 1491
  - Change Management 1491
  - Changes 1491
  - Color Code Status 1477
  - Connect On Diagram 1480
  - Connect Through Relationship Matrix 1480
  - Connectors 1480
  - Convert From Responsibility 1486
  - Create From Text File Text 1473
  - Create In Diagram 1473
  - Create In Project Browser 1473
  - Dependency Report 1492
  - Docked Windows 1488
  - Element 1471, 1483
  - Element Template 1478
  - Element, Hide/Show Connectors 955
  - Elements And Connectors 717
  - Extend Properties, Default Format 1478
  - External 850, 1483
  - Fast Generate Realization Connector 1480
  - Field Substitution Macros 1435
  - Functional 848
  - Gather 1468
  - Group, Toolbox 717
  - Hide Stereotype Letter 1471, 1473
  - Hierarchies 1483
  - Implementation Report 1492
  - Import Via CSV 1481

- Requirement
  - Inherited, Show 745
  - Internal 848, 1486
  - Internal, Import As Test 2227
  - Internal, In Scenarios & Requirements Window 879
  - Issues 1491
  - Manage 1488
  - Model 1468, 1483
  - Model Template 662, 666
  - Model Views 1491
  - Modeling 662, 666
  - Non-Functional 848
  - Profile 1478
  - Properties, Display On Diagram 1479
  - Properties, Extended 1478
  - Properties, Standard 1475
  - Realization 1480
  - Report Template 1492
  - Reporting 1492
  - Requirements Management 662, 666
  - Review 1488
  - Show Stereotype Letter 1471, 1473
  - Stability 848
  - Status, Color Coded 1477
  - SysML Model 2018
  - Tagged Values 1478
  - Tagged Values, Display 1479
  - Template 662, 666
  - Trace Through Connectors 1480
  - Trace Use Of 1490
  - User-Defined Attributes 1478
  - View 1489
  - What Is A? 1468
  - Windows For Tracing Use 1490
  - Windows For Viewing 1489
- Requirement Type
  - Define 998
- Requirements Diagram
  - Description 1494
  - Elements And Connectors 1494
  - Example 1494
- Requirements Management
  - And Enterprise Architect 1468
  - Enterprise Architect 9
  - In Example Model 1468
  - Overview 1468
- Requirements Model
  - SysML 2018
- Requirements Modeling
  - Enterprise Architect 9
- Reserved Names
  - In Shape Scripts, Connectors 1377
  - In Shape Scripts, Elements 1377
- Reset
  - Auto Increment Column Values 536
- Reset Options
  - For A Class 1922
  - For All Classes 1922
  - Source Code Language 583, 1922
- Reset Sort Order
  - In Project Browser 57
- Resize
  - Element 810
  - Element By Increments 810
  - Multiple Elements 810
- Resolve Change Conflicts
  - Between Replicas 250
- Resource
  - Add To Team Review Item 280
  - Allocation 450
  - And Tasking Details Dialog 477
  - Automation Interface, Element Package 2497
  - Delete From Team Review Item 280
  - Field Substitution Macros 1436
  - On Task Element 1676
  - Report 477
- Resource Allocation
  - In Project Calendar 510
  - Project Task Allocation Window, Resource View 473
  - Record Work, Personal Information Window 493
- Resource Description Framework 1643
  - Definition Diagram, Example 1651
- Resource Document
  - RTF Generator (Enhanced) 2259
  - RTF, Batch Generate Reports 2259
- Resource Management 449
  - Effort Types 456
  - Metric Types 457
  - Risk Types 458
- Resource View
  - Project Task Allocation 473
- Resources
  - Define 992
  - Documents 1003
  - Favorites Folder 1003, 1005
  - Matrix Profiles 1003
  - MDG Technologies 1003
  - Project Management Window 448
  - Scripts 1003
  - Templates 1003
  - UML Patterns 1003

- Resources
  - UML Profiles 1003
  - Window 1003
  - Window, And MDG Technologies 1284
  - XSL Stylesheets 1003
- Resources View
  - Of Patterns 1271
- Responsibility
  - Compartment, Element 842
  - Define 1486
  - Import As Test 2227
  - Internal 848
  - Move To External Requirement 1486
- Result
  - End Event, BPEL 1.1 1586
  - End Event, BPEL 2.0 1608
- Resynchronize
  - Package Version Control Status 380
- Reusable Subsystems
  - Systems Engineering Modeling 2023
- Reverse Connector 958
- Reverse Engineer
  - Enterprise Architect 9
  - Source Code 1790
  - Supported Languages 1790
- Reverse Engineering
  - And Auditing 393
  - And MDG Integration 1740
  - Binary Module, Supported Languages 1797
  - Directory Structure 1795
  - Eclipse 1740
  - Handling Classes Not Found During Import 1798
  - Import ActionScript 1793
  - Import Binary Module 1797
  - Import C 1793
  - Import C# 1793
  - Import C++ 1793
  - Import Delphi 1793
  - Import Java 1793
  - Import PHP 1793
  - Import Python 1793
  - Import Source Code 1792, 1798
  - Import Visual Basic 1793
  - Import Visual Basic.Net 1793
  - Initial Code In Operations 906
  - Introduction 1733
  - Languages Supported 1907
  - ODBC Data Sources 1973
  - Source Code, Import Directory Structure 1795
  - Supported Software Languages 1793
  - Synchronize Model And Code 1776
  - Visual Studio 1740
- Reverse Synchronization
  - Delete Attribute If Not In Code 1899
  - Delete Method If Not In Code 1899
  - Delete Model Aggregations For Attributes Not In Code 1899
  - Delete Model Associations For Attributes Not In Code 1899
  - Include Method Bodies In Model 1899
  - Of Package 1774
- Review
  - Package Version Control History (SCC) 377
  - Package Version Control History (TFS, CVS, Subversion) 376
  - Requirements 1488
- Rhapsody
  - Import Model From 420
- Rich Text Format
  - Copy Bookmark To Clipboard 575
  - Document 2251
  - Report 2251
- Rich Text Format Generator
  - Enhanced 2256
- Rich Text Format Report
  - Apply Filter (Legacy) 2320
  - Diagram Format (Legacy) 2321
  - Diagram Only 2340
  - Dialog (Legacy) 2318
  - Element-Level 2253
  - Exclude Elements (Legacy) 2321
  - Exclude Objects (Legacy) 2321
  - Exclude Package 2260
  - Generate (Enhanced) 2253
  - Generate (Legacy) 2323
  - Generate From Diagram List 596
  - Generate From Package Browser 587
  - Generate, Quick Start 2253
  - Generator (Legacy) 2318
  - Include Glossary (Legacy) 2321
  - Include Issues (Legacy) 2321
  - Include Package 2260
  - Include Tasks (Legacy) 2321
  - Object Selections (Legacy) 2323
  - Options (Legacy) 2322
  - Save As RTF Document (Enhanced) 2259
  - Save As RTF Document (Legacy) 2325
  - Set Main Properties (Legacy) 2319
  - Single Element (Legacy) 2319
  - Templates, Load (Legacy) 2323
  - Through Diagram List 2253
  - Through Model Search 2253
  - Through Package Browser 2253

- Rich Text Format Report
  - Wizard (Legacy) 2318
- Rich Text Notes
  - In Legacy RTF Generator Reports 2318
- Rigorous Security Mode
  - Apply User Lock 274
  - Release User Lock 274
- Risk
  - Automation Interface, Element Package 2498
  - Element 1675
  - Field Substitution Macros 1436
  - Management 453, 1675
  - Project Management Window 448
- Risk Types
  - Define 458
  - Global 458
  - Non-Global 453
- Roadmap
  - ICONIX 1925
- Robustness Diagram 1663
  - Generate From Scenario 864, 871
- Role
  - Context Menu 936
  - Define 991
  - Tagged Values 967
- Role Binding
  - Connector 1255
  - Relationship 1255
- RoleTag
  - Automation Interface, Connector Package 2520
- Rollback Change
  - Options 408
- Round-Trip Engineering
  - Introduction 1733
- RSA
  - Models, Import 419
  - XMI 417
- RSM
  - Import \*.emx Files 417
  - Import \*.uml2 Files 417
- RTF
  - Copy Bookmark To Clipboard 575, 581
  - Documentation 2249, 2251
  - Documentation, Other 2339
  - List Overrides 2299
  - Numbered Sections 2299
  - Numbering Levels 2299
  - Numbering List 2299
  - Report 2249, 2251
  - Section Numbering 2299
- RTF Document Options
  - RTF Generator (Enhanced) 727
- RTF Generator (Legacy) 727
- RTF Generator
  - Document Options (Enhanced) 727
  - Document Options (Legacy) 727
  - Document Options, From Diagram (Enhanced) 2255
  - Document Options, From Diagram (Legacy) 2255
  - Enhanced 2256
- RTF Report
  - Advanced Options 2261
  - Apply Filter (Legacy) 2320
  - Bookmarks 2346
  - Bookmarks, In Master Document Elements 2327
  - Custom Language Settings 2269
  - Custom Language Settings (Legacy) 2326
  - Diagram Format (Legacy) 2321
  - Diagram Only 2340
  - Dialog (Legacy) 2318
  - Document Options (Enhanced Generator) 2261
  - Element Filters 2264
  - Element-Level 2253
  - Embedded Elements, Include 2308
  - Exclude Elements (Legacy) 2321
  - Exclude Objects (Legacy) 2321
  - Exclude Package 2260
  - Footers, Add 2349
  - Generate (Enhanced) 2253
  - Generate (Legacy) 2318, 2323
  - Generate From Diagram List 596
  - Generate From Package Browser 587
  - Generate, Quick Start 2253
  - Headers, Add 2349
  - In MS Open Office 2261
  - In PDF Format 2256
  - In Sun Open Office 2256
  - Include Glossary (Legacy) 2321
  - Include Issues (Legacy) 2321
  - Include Package 2260
  - Include Tagged Values 2314
  - Include Tasks (Legacy) 2321
  - Keywords (Legacy) 2326
  - Linked Document, Render 2309
  - Object Selections (Legacy) 2323
  - Open Office Display 2261
  - Options (Legacy) 2322
  - Other Filters 2267
  - Project Constants, User Defined 2268
  - Quick Start 2253
  - Relationship Matrix Profile, Render 2310
  - Report External Elements 2310, 2312



- RTF Report
  - Save As RTF Document (Enhanced) 2259
  - Save As RTF Document (Legacy) 2325
  - Set Main Properties (Legacy) 2319
  - Single Element (Legacy) 2319
  - Switch Generator 2261
  - Templates, Load (Legacy) 2323
  - Through Diagram List 2253
  - Through Model Search 2253
  - Through Package Browser 2253
  - Update Links In MS Word 2351
  - Wizard (Legacy) 2318
  - Word Substitution 2269
- RTF Report Template
  - Add To MDG Technology 1334
- RTF Style Editor (Legacy) 2323
- RTF Style Template Editor
  - Add Content 2315
  - Bookmarks 2291
  - Character Formatting 2284
  - Character Styles 2284
  - Child Sections 2307
  - Columns 2296
  - Continuous Table 2305
  - Copy Text 2279
  - Cut And Paste 2279
  - Delete Text 2279
  - Drawing Objects 2302
  - Edit Picture 2282
  - Embedded Elements Sections 2308
  - Export As RTF Document 2277
  - File Options 2277
  - Fonts 2284
  - Footers 2290
  - Frames 2302
  - Headers 2290
  - Highlight Text 2279
  - Hyperlinks 2291
  - IME Option 2297
  - Import RTF Document 2277
  - Insert RTF File 2282
  - Level Numbering 2297
  - Lists And Overrides 2297
  - Move Text 2279
  - Non-rendered Subsections 2307
  - Numbered Lists 2285
  - Page Breaks 2289
  - Page Columns 2296
  - Page Mode 2281
  - Paragraph Formatting 2285
  - Paragraph Numbering 2285
  - Paragraph Styles 2285
- Paste External Objects 2279
- Picture Embed 2282
- Picture Frame 2302
- Picture Link 2282
- Print Options 2277
- Rendered Subsections 2307
- Repagination 2289
- Revert To Previous Copy 2277
- Save File 2277
- Search and Replace Options 2303
- Sections 2296
- Select Model Elements 2304
- Select Text 2279
- Show Headers & Footers 2281
- Special Text 2297
- Split Rows 2305
- Style Sheets 2297
- Tab Support (Enhanced) 2288
- Table Commands 2292
- Table of Contents 2297
- Tabular Sections 2305
- Tagged Value Section 2314
- Text Frame 2302
- Text Scrolling 2276
- Update Styles 2277
- ValueOf Field 2314
- View Options 2281
- Zoom 2281
- RTF Template
  - Create 2270
  - Delete 2270
  - Edit 2270
  - Export To Reference File 2270
  - Import 2316
  - Import From Reference File 2270
  - Normal.rtf 2270
- RTF Templates Tab (Enhanced) 2270
- Rule Composer
  - Business Rule Modeling 1536
  - Computation Rule Table 1536
  - Decision Table 1536
  - Rule Table 1536
- Rule Flow
  - Activity 1530
  - Activity Parameter 1533
  - Behavior 1530
  - Diagram 1530
  - Generate Code From Behavior 1543
  - Model 1530
- Rule Model
  - Diagram 1527
- Rule Task

- Rule Task
    - Element 1530
  - RuleFlow Diagram
    - Generate From Scenario 864, 867
  - Run
    - Application, Analyzer menu Option 120
    - SQL Patches 539
  - Run Command
    - Create 1862
  - Run Script
    - Create 1862
  - Run State 1197
    - Add Instance Variable 1198
    - Define, Element Context Menu 829
  - Run-Time
    - Variable, Define 1198
  - Runtime Object
    - In Script Editor 2378
  - Run-Time State
    - Add Instance Variable 1198
    - Delete Instance Variable 1198
    - Introduction 1197
- S -**
- Sample Model
    - Java 2145
    - Microsoft C++ 2145
    - Microsoft.NET 2145
    - Visual Execution Analyzer 2145
  - Save 428
    - Changes 67
    - Controlled Package 428
    - Diagram As UML Pattern 1271
    - Diagram Changes Automatically 553
    - Diagram Image To Disk File 741
    - Diagram, Context Menu Option 687
    - Diagram, Quick Start 67
    - Learning Center Profile 1354
    - Package with XMI 428
    - Profile From Diagram Context 1307
    - Profile From Package Context 1307
    - RTF Report As RTF Document (Enhanced) 2259
    - RTF Report As RTF Document (Legacy) 2325
    - UML Pattern 1271
  - Save As
    - Shortcut 183
  - Save Project As
    - Menu Option (File Menu) 92
  - Save Project As (File Menu Option)
    - Copy 185, 187
    - Shortcut (Capture Current Environment) 187
    - Shortcut (Direct Definition) 185
  - Scale
    - Image To Page Size 766
  - Scale Diagram View 550
  - Scan XML and Reconcile
    - Version Control 357
  - SCC
    - Package Version Control History, In 377
    - Providers Dialog 348
    - Version Control Options 348
    - Version Control, Upgrade For Enterprise Architect 4.5 344
  - Scenario
    - Alternate Path 852, 854
    - Alternate Path, Add 857, 859
    - Automation Interface, Element Package 2499
    - Basic Path 852
    - Context Menu, Item 859
    - Create 855, 857, 859
    - Create Element 861
    - Delete Path 863
    - Delete Step 857, 859
    - Delete Text 861
    - Description Tab 852
    - Edit Text 861
    - Element 852
    - Entry Points Tab 854
    - Entry Points Tab, Context Menu 863
    - Exception Path 852
    - Exception Path, 854
    - Exception Path, Add 857, 859
    - External Test Cases 873
    - Field Substitution Macros 1437
    - Floating Toolbar 864
    - Generate Activity Diagram From 865
    - Generate Diagram From 864
    - Generate From Activity Diagram 873
    - Generate From Clipboard Text 859
    - Generate Robustness Diagram From 871
    - Generate RuleFlow Diagram From 867
    - Generate Sequence Diagram From 869
    - Generate State Machine Diagram From 867
    - Generate Structure From Notes 852
    - Glossary Reference, Add 861
    - In Scenarios & Requirements Window 879
    - Insert Context Reference 861
    - Internal Test Cases 873
    - Item Context Menu 859
    - Join Scenarios 863
    - Link Step To Use Case 859, 861
    - Link To Element 861

- Scenario
  - Merge Steps 859
  - Move Step 857
  - Move Steps 859
  - Object 852
  - Organization 863
  - Set Responsible Entity 859
  - Split Step 861
  - Step Numbers On Sequence Messages 869
  - Structured Specification Tab 852, 854
  - Structured Specification, Create 855
  - Test Cases 873
  - Testing 2221
  - Text Context Menu 861
  - Toolbar 857
  - Type, Define 999
  - Undo Changes 861
  - Use Case 1078
- ScenarioDiagramType Enum
  - Automation Interface 2422
- ScenarioExtension
  - Automation Interface, Element Package 2501
- Scenarios & Requirements
  - Toggle Window/View 879
  - Window 879
- ScenarioStep
  - Automation Interface, Element Package 2502
- ScenarioStepType Enum
  - Automation Interface 2422
- ScenarioTestType Enum
  - Automation Interface 2423
- Schema
  - ArcGIS, Export 1640
  - ArcGIS, Import 1641
  - Database 1631
  - Database, Import From ODBC 1973
  - DDL, Import From ODBC 1973
  - Owner Tagged Value 1939
  - Set Owner 1939
- Scope
  - Values 1741
- Scope Guides
  - Code Editor, Common 1810
- Screen
  - Element 1656
- Script
  - Add To MDG Technology 1334
  - Commands 2376
  - Console 2376
  - Copy 2373
  - Create 2373
  - Delete 2373
  - Deploy, Create 1862
  - Diagram 2375
  - Engines 2372
  - Execute 2373
  - Group 2372, 2373
  - Group Type 2375
  - Hyperlink To 1670
  - JavaScript 2372, 2373
  - JScript 2372, 2373
  - Local 2372, 2373
  - Model Search 614, 2375
  - Move 2373
  - Normal 2375
  - Project Browser 2375
  - Search 611, 1817, 2375
  - Template 2372, 2373
  - Unit Test, Create 1834
  - Unit Test, Introduction 1834
  - User 2372, 2373
  - VBScript 2372, 2373
  - Workflow 2373, 2375
  - Workflow Functions 299
  - Workflow, Introduction 298
- Script Actions
  - Define, Visual Execution Analyzer 1831
- Script Editor
  - Context Menu 1805
- Script Objects 2378
- Scripting
  - In Enterprise Architect 2371
- Scripting Window
  - Console Tab 2376
  - Console Toolbar 2376
  - Context Menu 2373
  - Create Script 2372
  - Editor 2378
  - Group Properties Dialog 2375
  - Intellisense 2378
  - Runtime Objects 2378
  - Script Editor 2378
  - Script Objects 2378
  - Scripts Tab 2373
  - Toolbar 2373
  - Type Libraries 2378
- Scripts
  - Build 1832
- SDK
  - Enterprise Architect 1292
- Search
  - Add-In 2594
  - Advanced Options 626
  - Code Editor Facility 1805

- Search
  - Conditions 629
  - Configure Code Editor Context Menu Options 1805
  - Debugger File Search 1817
  - Definitions 621
  - Definitions, New 623
  - Definitions, Predefined 617
  - Element Features 621
  - Element Features, RTF Reports 2267
  - Element Filters, RTF Reports 2264
  - Fields 629
  - Filter On Feature Values 626
  - Filters 621
  - List, Predefined Searches 617
  - Macros, SQL 623
  - Manage 621
  - Model 621
  - Model, Create Search Definition 623
  - Other Filters, RTF Reports 2267
  - Project 621
  - Query, New 623
  - Results, Manipulate 611
  - Scripts 1817, 2375
  - Simple 621
  - Specific Package 626
  - Team Review 295
  - User Defined, Storage 623
  - Whole Model 626
  - Within Project Browser 583
- Search Data Parameter
  - Add-In Search 2595
- Search Filters
  - Add To Search 627
- Search Project
  - Add Filters 627
- Search Tools 566
- Section Numbering
  - In RTF Reports 2299
  - In Virtual Documents 2337
- Security
  - Basics 251
  - Change Password 256, 270
  - Disable 253
  - Enable 253
  - Maintain Groups 262
  - Maintain Users 256
  - Policy 255
  - Re-enable 253
  - Require User Lock Mode 255
  - Reset Password 270
  - Rigorous Security 255
  - Set Password 270
  - Standard Security 255
  - Submenu (Project Menu) 104
  - Tasks 251
  - User Permission List 264
  - User/Group Lock Mode 255
  - What Is User Security? 251
- Security Group Permissions 263
- See the Pass Parameters to Rule Flow Activity 1530
- Select
  - Alternative Image 755
  - ODBC Data Source 1975
  - Stereotypes 1266
- Select <Item> Dialog 880
- Select All
  - Menu Option (Edit Menu) 93
- Select Attribute Type
  - Data Type 887
  - Dialog 887
- Select By Type
  - Menu Option (Edit Menu) 93
- Select Property Dialog 882
- SELECT Statement
  - SQL Search 623
- Selectable
  - Element 837
- Self Message
  - Calls 1236
- Self-Message
  - Connector 1235
  - Hierarchy, Sequence Diagram 1082
  - Relationship 1235
  - Return 1235
- Send
  - Element 1151
  - Event 1667
- SendSignal Action
  - Signal Tab 1095
- Sent Mail
  - Personal Information Window 487
- Sequence
  - Communication Messages 1244
  - Elements and Connectors 710
  - Message, Change Timing Details 1238
  - Message, Create 1232
  - Message, Timing Details 1238
  - Oracle, DDL Options 1978
  - Oracle, DDL Options For Packages 1979
- Sequence Diagram
  - Activation Levels 1082
  - And Version Control 1079

- Sequence Diagram
  - Damage To 1079
  - Description 1074
  - Diagram Features, Generate Sequence Diagrams 2152
  - Display Options 555
  - Element Activation 1080
  - Elements 1078
  - Elements And Connectors 1074
  - Example 1076
  - From Code Execution, Control Stack Depth 2153
  - Generate Code From 1778, 1786
  - Generate From Debugger Call Stack 1886
  - Generate From Recording, Execution Analysis 2164
  - Generate From Scenario 864, 869
  - Generate In Execution Analysis 2148
  - Generate, Execution Analysis 2164
  - Layout 1077
  - Lifeline Activation Level 1082
  - Messages, Asynchronous Signal 1241
  - Messages, Self Message 1235
  - Recording History, Execution Analysis 2149
  - Save Recording History, Execution Analysis 2164
  - Self-Message Hierarchy 1082
  - Step Numbers On Messages 869
  - Top Margin, Change 1083
- Sequence Diagram Message
  - Examples 1237
  - External To Sequence 1237
  - General Ordering 1240
- Sequence Diagrams
  - Filter calls 1863
- Sequence Element
  - Inline, Part And Port 1084
- Sequence Flow
  - Create In BPEL 1.1 1587
  - Create In BPEL 2.0 1612
  - Model In BPEL 1.1 1587
  - Model In BPEL 2.0 1612
- Sequence Message
  - Label Visibility 1083
  - Modify Height 1077
- Sequence Recording Option
  - Record Activity For Class 2160
  - Record Activity For Method 2155
- Sequential Node
  - Structured Activity 1161, 1164
- Server
  - Apache Tomcat, Debugging 1846
  - JBOSS, Debugging 1846
  - Tomcat, Debugging 1846
- Server Configuration
  - JBOSS 1848
  - Tomcat 1849
- Server Repository
  - Create for Oracle 10g 197
  - Create for Oracle 9i 197
- Service
  - WSDL Diagram 2082
  - WSDL Element 2082
- Service Configuration
  - Tomcat 1850
- Service Oriented Architecture
  - Development 1736
  - Implementing XML-Based, In Enterprise Architect 1736
- Service Oriented Architecture Modeling Language 2088
- Service Oriented Modeling Framework 2092
- Set
  - Activities For Transitions 880
  - Association Specialization 958
  - Classifiers For REFGUID Tagged Values 880
  - Collection Classes 1904
  - Connector Visibility 953
  - Default Diagram, Model 744
  - Default Tree Behavior 587
  - Diagram Appearance Options 725
  - Diagram Page Size 764
  - Diagram Properties 725
  - Element Classifier 1274
  - Element Cross References 803
  - Element Custom References 803
  - Element Parent 801
  - Feature Visibility 745
  - Font For Element Text 839
  - Group Permissions 263
  - Instance Classifier 880
  - Interface For Element 801
  - Main RTF Report Properties (Legacy) 2319
  - Message Source and Target 959
  - Object State 1199
  - Operation Parameter Return Type 880
  - Operation Parameter Type 880
  - Operation Return Type 880
  - Parent For Element 801
  - Pattern Element Defaults 880
  - Relationship Visibility 953
- Set Attribute Dialog 884
- Set Coordinate System
  - ArcGIS 1637

- Set Element Project Colors 548
- Set Feature Dialog 884
- Set Feature Visibility
  - Element Context Menu Option 836
- Set Function
  - Create As Attribute Property 892
- Set Operation Dialog 884
- Set Project Custom Colors
  - For Element 819
- Set Up
  - Adaptive Server Anywhere ODBC Driver 205
  - Database Model Files 193
  - Debug Session 1837
  - For .NET 1850
  - MySQL ODBC Driver 201
  - ODBC Driver 200
  - Oracle ODBC Driver 208
  - PostgreSQL ODBC Driver 202
  - Progress OpenEdge ODBC Driver 206
  - Single Permissions 260
  - User Groups 260
- Settings
  - Author 989
  - Cardinality (Multiplicity) 986
  - Clients 993
  - Constraint Status Types 998
  - Constraint Types 997
  - Default Hours 520
  - Effort Types 456
  - Environment Complexity Factors 518
  - Estimation 1000
  - General Types 995
  - Maintenance 1001
  - Menu 981
  - Menu, Configure 127
  - Metric Types 457
  - Metrics 1000
  - People 989, 992
  - Problem Types 1001
  - Project Author 989
  - Project Resources 992
  - Requirement Types 998
  - Risk Types 458
  - Roles 991
  - Scenario Types 999
  - Status Types 995
  - Stereotypes 982
  - Tagged Value Types 985
  - Technical Complexity Factors 517
  - Template Package 820
  - Testing Types 1002
  - UML Types 981
- Shallow Copy
  - Of Diagram 742
- Shallow History 1134
  - Change To Deep 829
- Shape
  - <LabelID> 1377
  - Attributes 1363
  - Decoration 1377
  - Editor 1359
  - Label 1377
  - Main 1377
  - Source 1377
  - Target 1377
- Shape Attributes
  - Shape Scripts 1363
- Shape Editor 984, 1359
- Shape Script
  - Add Custom Compartments 1339
  - Block Layout Examples 1360
  - Order Of Declaration 1360
  - Overview 1360
  - Sub Shapes 1360
- Shape Scripts
  - Add To Profile 1304
  - Alter Embedded Element 1380
  - Arithmetical Operations 1378
  - Assign To Stereotype 1358
  - Basic Shapes 1380
  - Change Font Of Text 1378
  - Cloud Path 1380
  - Color Queries 1372
  - Comments 1378
  - Conditional Branching 1373
  - Connector 1380
  - Create 1358
  - Custom Shapes 1358
  - Display Element Properties 1373
  - Double Line 1380
  - Drawing Methods 1366
  - Editable Field 1380
  - Example Shape Scripts 1380
  - Filled Arrow 1380
  - Fonts 1378
  - Getting Started 1358
  - Introduction 1358
  - Looping 1378
  - Miscellaneous 1378
  - Multiple Condition 1380
  - Override Element Appearance 1358
  - Properties, Connector 1373
  - Properties, Element 1373
  - Query Methods 1373

- Shape Scripts
  - Reserved Names, Connectors 1377
  - Reserved Names, Elements 1377
  - Return Statement 1380
  - Shape Attributes 1363
  - Shape Editor 984, 1359
  - Single Condition 1380
  - Stereotypes 1358
  - String Manipulation 1378
  - Subshape 1380
  - Subshape Layout 1376
  - Syntax Grammar 1379
  - Terminate Execution 1378
  - Variable Declarations 1378
  - Without Stereotypes 1378
  - Writing Scripts 1363
- Share
  - An Enterprise Architect Project 242
  - Project On Network Drive 243
- Shared Key
  - Add 2739
  - Issues 2742
- Shared Model Development
  - Enterprise Architect 9
- Shortcut
  - Clear 187
  - Diagram (Capture Current Environment) 187
  - Keyboard 166
  - Keyboard/Mouse 174
  - Menu, Toolbox 704
  - Model (Capture Current Environment) 187
  - Model Search (Capture Current Environment) 187
  - Relationship Matrix (Capture Current Environment) 187
  - Team Review (Capture Current Environment) 187
- Shortcut To
  - Diagram 183
  - Diagram (Direct Definition) 185
  - Model 183
  - Model (Direct Definition) 185
  - Model Search 183
  - Model Search (Direct Definition) 185
  - Relationship Matrix 183
  - Relationship Matrix (Direct Definition) 185
  - Team Review 183
  - Team Review (Direct Definition) 185
- Show
  - Connectors, All Diagrams 955
  - Connectors, Requirements Element 955
  - Connectors, Single Diagram 955
  - Diagram Caption Bar 98
  - Duplicate Tagged Values 974
  - Element Stereotype 1266
  - Feature Stereotype 1266
  - Labels 957
  - Package Contents On Diagram 685
  - Project Browser 567
  - Relationship 851
  - Toolbox 698
  - Usage Of Element 802
  - Use Case Arrowhead 959
- Show Grid
  - Diagram Menu Option 105
- Show Status Colors on Diagrams
  - Menu Option 1477
- Show/Hide
  - Connectors 941
  - Labels 941
- Signal
  - Element 1205
  - Reception 1205
- Simple View 676
- Simulate
  - SysML Parametric Model 2016
- Simulate the parametric models to verify their 1984
- Simulation
  - Multi-Threaded 2137
- Simulation Events Tab
  - Context Menu Options 2125
  - Toolbar Option 2125
  - Use In Simulation 2125
- Simulation Script
  - Activate 2111
  - Evaluate Guards And Effects 2109
  - Record Simulation 2109
  - Run 2112
  - Set Up 2109
- Single Permissions
  - Set Up 260
- Single User 1901
- Size
  - Elements By Increments 112
  - Submenu 112
- Slideshow
  - Diagram Presentations 598, 607
  - Model View 598, 607
  - Of Diagrams, Automate 607
  - Properties Dialog, Model View 607
  - Remove From Model View 607
  - Run, Diagram View 607
  - Run, Full Screen View 607
  - Stop 607

- Snap to Grid
  - Diagram Menu Option 105
- Snapshot
  - Of Variables In Code Execution 1881
- SOA
  - Development 1736
  - Implementing XML-Based, In Enterprise Architect 1736
  - Modeling Language 2088
- SOA WSDL
  - Elements And Connectors, Enterprise Architect 719
- SOA XSD
  - Elements and Connectors, Enterprise Architect 720
- SoaML
  - Concept 2088
  - Diagrams 2088
  - Disable 2088
  - Elements 2088
  - Enable 2088
  - MDG Technology 2088
  - Toolbox Pages 2088
- SOAP Binding 2078
- Software and Systems Process Engineering Meta-model
  - SPEM 523
- Software Architect
  - And Enterprise Architect 75
  - Project Role 75
- Software Development
  - Database Engineering 1931
  - MDA Transformations 1679
  - Model Transformations 1679
  - Software Engineering 1733
  - XML Engineering 2026
- Software Development Kit
  - Enterprise Architect 1292
- Software Engineer
  - And Enterprise Architect 77
  - Project Role 77
- Software Product License Agreement 28
- SOMF
  - Concept 2092
  - Diagrams 2092
  - Disable 2092
  - Elements 2092
  - Enable 2092
  - MDG Technology 2092
  - Toolbox Pages 2092
- Sort Order
  - Project Browser, Reset 57
- Source
  - Set for Message 959
- Source Code
  - Add New Features And Elements 1404
  - Control 312
  - Display In Source Code Viewer 1799
  - Display Parameter Information 1802
  - Editor Functions 1802
  - Editors 1799
  - Engineering, Project Browser Options 583
  - File Parsing In Source Code Viewer 1799
  - Generate For Method In Project Browser 582
  - Generate For Operation In Project Browser 582
  - Import, Reverse Engineering 1792
  - Internal Editor Options 1895
  - Open Directory, Project Browser Menu Option 577
  - Record Macro 1802
  - Reset Language 583, 1922
  - Reverse Engineering, Supported Languages 1790
  - Synchronize 1790
  - Synchronize With Method In Project Browser 582
  - Synchronize With Operation In Project Browser 582
  - View For Method In Project Browser 582
  - View For Operation In Project Browser 582
  - View, Project Browser Menu Option 577
  - Viewer 1799
  - Viewer Toolbar 1802
  - XML Structure Tree 1799
- Source Code Editor
  - Compare 1801
  - External 1801
  - Internal (External Code) 1801
  - Variants 1801
  - View 1801
  - Window 1801
- Source Code Engineering
  - Submenu (Element Menu) 111
  - Submenu (Tools Menu) 116
- Source Code Generation
  - Class 1768
  - Interface 1768
  - Options, Field Substitution Macros 1418
  - Overview 1768
  - Supported Software Languages 1768
- Source File
  - Open (Option) 115
- Source Object
  - Multiplicity 964



- Source Role
  - Details 964
- Space Evenly
  - Submenu 112
- Sparx Systems
  - Enterprise Architect Community Site 40, 87
  - Website 40
- Spatial Reference
  - ArcGIS 1637
  - Class, Create 1637
  - Element, Create 1637
  - Package 1637
  - Tagged Value 1637
- Specialize Association 958
- Specialized UML Models 1498
- Spectacle Icon
  - Inside Element 825
- Spell Check
  - Correct Words 486
  - Dictionary 486
  - Languages Other Than English 481
  - Model 485
  - Options, Configure 481
  - Perform 485
  - Project 485
  - Single Package 485
- Spell Checking
  - Automatic, Disable 480
  - Automatic, Enable 480
  - Introduction 480
- Spelling Options
  - Select 481
- SPEM
  - Base Plug-In 523
  - Concept 523
  - Diagram 524
  - Disable 523
  - Elements 524
  - Enable 523
  - MDG Technology 523
  - Method Content 523
  - Package 523
  - Process 523
  - Software and Systems Process Engineering Meta-model 523
  - Stereotype Presentation 524
  - Toolbox Page 524
- Spring Diagram Layout 780
- SQL
  - Custom Searches 623
  - Editor 623
  - Patches, Run 539
  - Search Macros 623
  - Search, SELECT Statements 623
  - Server Data Repository, Connect To 212
  - Server Repository, Create 195
- SQL Server
  - Default Constraint 1971
  - Desktop Engine, Upsize To 236
  - Non-Clustered Primary Key 1955
  - Upsize To 228
- Stack
  - Depth, Control 2153
- Standard Colors
  - Attribute 546
  - Connector Line 546
  - Diagram Background 546
  - Element Fill 546
  - Element Line 546
  - Method 546
  - Note Text 546
  - Operation 546
  - Options 546
  - Screen 546
  - Set Element Project Colors 548
  - Shadow 546
- Standard Element Stereotypes 1267
- Start
  - Application 47
  - Enterprise Architect 47
- Start Event, BPEL 1.1
  - Create 1577
  - Model 1577
  - Types 1577
- Start Event, BPEL 2.0
  - Create 1601
  - Model 1601
  - Types 1601
- Start Page 87
  - Hide 129, 695
  - Quick Start 49
  - Show 129
  - Show When Hidden 691
- State
  - Chart 1032
  - Composite 1152, 1153
  - Diagram 1032
  - Element 1152
  - Entry And Exit Actions 901, 1152
  - In Timeline Element 1062
  - Locate In State Machine Diagram 1047
  - Locate In State Machine Table 1047
  - Operation Behavior 904
  - Reposition In State Machine Table 1046

- State
  - Simple 1152
  - State Machine Table Conventions 1047
- State (Machine)
  - Elements and Connectors 711
  - Group, Toolbox 711
- State Changes
  - Capture, Execution Analysis 2166
  - Map, Visual Execution Analyzer 2168
  - Record, Visual Execution Analyzer 2168
  - Set Up To Capture, Execution Analysis 2166
- State Invariant
  - Element 1155, 1157
- State Lifeline
  - Element 1158
- State Lifeline Element
  - Add State 1055
  - Add To Timing Diagram 1054
  - Add Transition 1058
  - Change Transition Time 1057
  - Define Name 1054
  - Delete State 1057
  - Delete Transition 1057
  - Edit State 1056
  - Edit Transition 1057
  - Merge Transitions 1057
  - Move Transition 1058
  - Set Timeline Start Position 1054
  - Sizing and Scale 1054
  - Synchronize Transition 1057
- State Machine
  - Element 1161
  - Entry And Exit Actions 901
  - Execution Analysis 2166
  - In Visual Execution Analyzer 2166
  - Include in Execution Analysis Recording 2161
  - Model For Hardware Description Languages 1784
  - Regions 1037
- State Machine Diagram
  - Code Generated From 1781
  - Description 1032
  - Display Format 1032
  - Elements And Connectors 1032
  - Example 1034
  - Generate Code From 1778, 1780
  - Generate From Scenario 864, 867
  - Locate State In State Machine Table 1047
  - Locate Transition In State Machine Table 1047
  - Locate Trigger In State Machine Table 1047
- State Machine Table
  - Add States 1043
  - Add Substates 1043
  - Add Triggers 1044
  - Cell Color 1039
  - Cell Enumeration 1039
  - Cell Highlights 1039
  - Cell Size 1039
  - Change Position In Diagram View 1042
  - Change Size 1043
  - ChangeTransitions 1045
  - Conventions 1047
  - Description 1038
  - Export To CSV 1048
  - Format 1038
  - Insert Transitions 1045
  - Legend, Add 1046
  - Legend, Remove 1046
  - Locate State In State Machine Diagram 1047
  - Locate Transition In State Machine Diagram 1047
  - Locate Trigger In State Machine Diagram 1047
  - Operations, Overview 1042
  - Options 1039
  - Remove Substate Parent Relation 1043
  - Reposition States 1046
  - Reposition Substates 1046
  - Reposition Triggers 1046
  - State-Next State 1038, 1050
  - State-Trigger 1038, 1049
  - Table Format 1039
  - Trigger-State 1038
- State Region
  - Composite 1037
- State Transition
  - Record In Execution Analysis Recording 2161
- State Transitions
  - Record, Visual Execution Analyzer 2165
- State/Continuation
  - Element 1155
- Statement
  - Trace 1873
- Status Bar
  - Element Coordinates 152
  - Enterprise Architect Workspace 152
  - Hide 152
  - Show 152
  - Zoom Control 152
- Status Type
  - Color 995
  - Define 995
  - For Different Elements 995
- Step Into
  - Function Calls 1875

- Step Out Of
  - Functions 1875
- Step Over
  - Lines Of Code 1875
- Step Through
  - Function Calls 2162
- Stereotype
  - Add Shape Script In Profile 1304
  - Add To Profile 1297
  - Add, Automation Interface Code Example 2580
  - Analysis 1508
  - And Metafiles 1264
  - Apply To Dependency Relationship 1221
  - Automation Interface Repository 2470
  - Custom 1293
  - Define As Metatype 1312
  - Definition 1264
  - Dialog 1293
  - Extension 1663
  - Inbuilt 1663
  - Multiple, Restrict Application Of 1313
  - Predefined Tag Types 1300
  - Profile 1316
  - Selector 1266
  - Set Default Appearance Of Objects In Profile 1306
  - Settings 982
  - Show On Project Browser 587
  - SPEM Presentation 524
  - Standard Element 1267
  - Synchronize Element With Profile 1281
  - Tagged Values In Profile 1299
  - Tags For Supported Attributes 1301
  - Tags, Define 1299
  - UML Description 1264
  - Visibility 1266
  - With Alternative Images 1269
  - XSD In UML Profile 2055
- Stereotyped Element
  - Table 1630
- Store
  - Image In Enterprise Architect 755
- Stored Procedure
  - As Individual Class 1961
  - Definition 1961
  - Element 1961
  - Select From ODBC Data Source 1976
  - Supported Databases 1961
- Straighten
  - Line At Cursor 948
- Strategic Modeling
  - Integrated Technology 1289
- String Viewer Dialog 1879
- Structural Diagram
  - Elements 1179
- Structural Diagrams
  - Overview 1011
- Structural Specification
  - Generate From Description 852
- StructuralFeature Action
  - Set Structural Feature 1095
- Structured Activity
  - Conditional Node 1161, 1168
  - Element 1161, 1163, 1164, 1168
  - Loop Node 1161, 1164
  - Nested 1161
  - Node 1161, 1163
  - Sequential Node 1161, 1164
- Structured Activity Node 1163
- Structured Scenario
  - Constraints Tab 875
- Structured Specification
  - Alternate Path 854
  - Alternate Path, Add 857, 859
  - Context Menu, Item 859
  - Create 855, 857, 859
  - Create Element 861
  - Delete Path 863
  - Delete Step 857, 859
  - Delete Text 861
  - Edit Text 861
  - Entry Points Tab 854
  - Entry Points Tab, Context Menu 863
  - Exception Path 854
  - Exception Path, Add 857, 859
  - External Test Cases 873
  - Floating Toolbar 864
  - Generate From Activity Diagram 873
  - Generate From Clipboard Text 859
  - Glossary Reference, Add 861
  - Insert Context Reference 861
  - Internal Test Cases 873
  - Item Context Menu 859
  - Join Scenarios 863
  - Link Step To Use Case 859, 861
  - Link To Element 861
  - Merge Steps 859
  - Move Steps 859
  - Of Scenario Steps 854, 855
  - Organization 863
  - Set Responsible Entity 859
  - Split Step 861
  - Test Cases 873
  - Text Context Menu 861

- Structured Specification
  - Toolbar 857
  - Undo Changes 861
- Structured Tagged Value Type 1386
- Structured Tagged Values
  - Create 1389
- Style
  - For Connectors 940
- Style Template
  - Fragments, HTML 2357
- StyleEx
  - Diagram Profile Attribute Values 1348
- Sub Activity
  - As Hyperlink 1670
- Sub-Activity
  - Conditional Node 1161, 1168
  - Element 1101, 1161, 1163
  - Loop Node 1161, 1164
- Submachine State
  - Element 1152, 1161
- Submenu
  - Add-In 125
  - Advanced (Element) 109
  - Alignment 112
  - Appearance (Element) 112
  - Data Management (Tools Menu) 119
  - Database Engineering (Tools Menu) 117
  - Debug (Analyzer) 122
  - Documentation (Project Menu) 101
  - Hidden, Create In Toolbox Profile 1341
  - Import/Export (Project Menu) 102
  - Inline Features (Element Menu) 110
  - Make Same 112
  - Manage .EAP File (Data Management Menu) 120
  - Model Transformations (Tools Menu) 118
  - Model Validation (Project Menu) 102
  - More Analyzer Windows 125
  - More Element Tools 98
  - More Project Tools 98
  - Move 112
  - Online Resources 131
  - Paste Elements (Edit Menu) 94
  - Project Types 129
  - Record (Analyzer) 123
  - Security (Project Menu) 104
  - Size 112
  - Source Code Engineering (Element Menu) 111
  - Source Code Engineering (Tools Menu) 116
  - Space Evenly 112
  - Test (Analyzer) 124
  - Toolbars 98
  - Version Control (Project Menu) 103
  - Visual Styles 98
  - Web Services (Tools Menu) 118
  - XML Schema (Tools Menu) 118
  - Zoom 763
  - Z-Order 112
- Subsetted
  - Property, Port 1204
- Subsetted Property
  - Of Attribute 889
- Subshape
  - Example 1376
  - In Shape Scripts 1376
- Substate
  - Reposition In State Machine Table 1046
- Sub-State 1153
- Substitute Words
  - In Extended RTF Generator 2269
  - In RTF Report (Legacy) 2326
- Substitution
  - Conditional 1413
  - Direct 1413
  - Macro 1373
- Subtype
  - Relationship 955
- Subversion
  - Package Version Control History, In 376
  - TortoiseSVN 334
  - Using With Enterprise Architect Under WINE Crossover 332
  - Version Control Options 351
  - Version Control, Create Local Working Copy 330
  - Version Control, Create Repository Subtree 329
  - Windows-Based Client 332
- Support
  - For Registered Users 42
  - For Trial Users 42
- Supported Attribute
  - Create Composite Elements 1314
  - Define Behavior On Creating Instance 1313
  - Define Child Diagram Types 1315
  - Metatype, In Profiles 1308
  - Of Stereotype Tags 1301
  - Stereotype, In Profiles 1308
- Supported DBMSs 1970
- Supported Languages
  - Code Engineering 1907
  - Code Generation 1907
  - Code Generation From Behavioral Models, Software Languages 1778
  - Code Generation, Hardware Definition 1741

- Supported Languages
  - Code Generation, Hardware Description Languages 1778
  - Code Generation, Software 1741
  - Debugging 1734
  - Execution Analysis 1734
  - Hardware Description Languages 1784
  - Hardware Description Languages, Code Generation 1778
  - Recording 1734
  - Reverse Engineering 1790, 1907
  - Software Code Generation 1768
  - Software, Import 1793
  - Software, Reverse Engineering 1793
- Supported Stereotype Attribute Tags 1301
- Suppress
  - Line Segments 948
- SwimlaneDef
  - Automation Interface, Diagram Package 2531
- Swimlanes
  - Automation Interface, Diagram Package 2532, 2534
  - Manage 751
  - On Diagram 751
  - Orientation 751
- Swimlanes Matrix
  - Activate 753
  - And Matrix Dialog, Matrix Tab 753
  - Create Columns And Rows 753
  - Define Heading 753
  - Delete Items 753
  - Edit Items 753
  - Lock 753
  - Model Profile 753
  - Size 753
- Switch
  - Between Diagram Tabs 697
  - Diagrams 691
  - RTF Generator 2261
- Sybase Adaptive Server Anywhere
  - ODBC Driver, Set Up 205
  - Upsize To 234
- Synch
  - Element 1169
- Synchronization 244
  - Initial Code In Operations 906
  - Introduction 1733
  - Macros, Code Template Syntax 1447
  - Of Source Code And Model 1790
- Synchronize
  - Batch With Code 111
  - By Dragging Element From Toolbox 1281
  - Call Argument With Behavior Parameter 911
  - Class With Code 111
  - Classes, Import Database Schema 1973
  - Code 1401
  - Elements With Profile 1281
  - Existing Code Sections 1403
  - Invocation Argument With Behavior Parameter 911
  - Package With Source, (Tools Menu Option) 116
  - Replicas 247
  - Stereotypes From Profile 1281
  - Tagged Values And Constraints 1281
  - Tagged Values From MDG Toolbox Pages 1281
  - UML Profile Tagged Values And Constraints 1281
- Syntax Check
  - For UML, Option 549
- Syntax Grammar
  - Shape Scripts 1379
- Syntax Highlighting
  - Code Editor Options 1896
  - Code Editor, Common 1810
- SysML
  - Activity Elements, Toolbox Page 1992
  - Activity Relationships, Toolbox Page 1992
  - Block Definition Elements, Toolbox Page 1996
  - Block Relationships, Toolbox Page 1996
  - Concepts 1990
  - Design Model 2021
  - Diagrams 1990
  - Disable 1990
  - Interaction Elements, Toolbox Page 1999
  - Interaction Relationships, Toolbox Page 1999
  - Internal Block Diagram 2021
  - Internal Block Elements, Toolbox Page 2000
  - Internal Block Relationships, Toolbox Page 2000
  - MDG Technology 1990
  - MDG Technology For, Enterprise Architect 1289
  - Migrate 1.1 Model To 1.2 2024
  - Model Elements, Toolbox Page 2002
  - Model Relationships, Toolbox Page 2002
  - Operational Domain Model 2019
  - Parametric Elements, Toolbox Page 2005
  - Parametric Model, Simulate 2016
  - Parametric Models 2013
  - Parametric Relationships, Toolbox Page 2005
  - Requirement Elements, Toolbox Page 2007
  - Requirement Extensions, Toolbox Page 2007

## SysML

- Requirement Relationships, Toolbox Page 2007
- Requirements Model 2018
- Reusable Subsystems 2023
- State Machine Elements, Toolbox Page 2009
- State Machine Relationships, Toolbox Page 2009
- Toolbox Pages 1990
- Use Case Elements, Toolbox Page 2011
- Use Case Relationships, Toolbox Page 2011

## System

- Project Glossary 460
- Project Issues 460
- Project Tasks 460
- Tabs 460
- Testing 2218
- Users 256
- Window 460

## System Boundary

- Element 1170
- Insert New From Toolbar 143

## System Design,

- Compose, Systems Engineering Modeling 2021

## System Engineering Modeling

- Create Reusable Subsystems 2023

## System Window

- Project Glossary Tab 468
- Project Issues Tab 462
- Project Tasks Tab 460

## SystemC

- Code Generation 1917
- Enterprise Architect Toolbox Pages 1760
- Language Options 1917
- Modeling Conventions 1760

## Systems Engineering Edition

- Of Enterprise Architect 22

## Systems Engineering Model

- Create 1984

## Systems Engineering Modeling

- Compose System Design 2021
- Create SysML Parametric Model 2013
- Overview 1984
- Process 1984
- Simulate SysML Parametric Model 2016
- SysLM Operational Domain Model 2019
- SysML Requirements Model 2018

## Systems Modeling Languages

- MDG Technology For, Enterprise Architect 1289

## Systems Modelling Language (SysML) 1990

## - T -

## Tab

- Asterisks 695
- Audit History In Output Window 164
- Close 695
- Context Menu 695
- Context References 874
- Diagram 695
- View 695

## Tab Support

- RTF Style Template Editor (Enhanced) 2288

## Tabbed Frame

- Combine Windows In 135
- Remove Window From 135

## Table

- Create 1936
- Data Modeling 1935
- Data Type Conversion 1946
- DDL Script For 1625
- Detail 1630
- Element 1630
- Introduction 1935
- Owner Tagged Value 1939
- Properties 1630
- Set Database Type 1938
- Set Owner 1939
- Set Properties 1937
- State Machine 1038

## Table Of Contents

- MS Word, Add 2348

## Table Of Figures

- MS Word, Add 2349

## Tables in RTF Reports

- Sections 2305
- Split Rows 2305

## Tag

- Compartment, Element 842
- Management, Advanced 975
- Profile 1299

## Tag Type

- Predefined, Assign To Stereotype 1300

## Tagged Region

- Code Editor Find And Replace 1814

## Tagged Value

- Add 971
- Add To Operations 907
- Assign Information To 973
- Assign To Item 972
- Behavior Parameters 914

- Tagged Value
  - By Dragging Element From Toolbox 1281
  - Connector, Use 1301
  - Custom, Create 1393
  - Duplicate Values 969
  - Element Package, Automation Interface 2504
  - Entity Relationship Diagram 1627
  - For Oracle Table Properties 1940
  - For Schema Owner 1939
  - For Table Owner 1939
  - Fully Qualified, Show 745
  - Fully-Qualified Value 969
  - In UML Profiles 1280
  - Inherited, Show 745
  - Insert Column In Package Browser 592
  - Macros, Code Template Syntax 1437
  - Masked, Create 1393
  - Model Components And 969
  - Modify 969
  - Modify Value 972
  - Of Attributes 891
  - Quick Add 971
  - Reference Data, Create 1392
  - Show Duplicates 974
  - Structured, Create 1389
  - Synchronize, And Constraints 1281
  - Synchronize, From MDG Toolbox Pages 1281
  - Toolbar 969
  - Types 985
  - Types Of Value Field 972
  - View 969
  - What Is A? 969
  - Window 969
- Tagged Value Broadcast Events
  - Add-In Model 2658
  - EA\_OnAttributeTagEdit 2659
  - EA\_OnConnectorTagEdit 2660
  - EA\_OnElementTagEdit 2661
  - EA\_OnMethodTagEdit 2662
- Tagged Value Section
  - RTF Style Template Editor 2314
- Tagged Value Type
  - Add To MDG Technology 1331
  - Filters 1386
  - Introduction 1385
  - Predefined Reference Data 1390
  - Predefined Structured 1386
- TaggedValue
  - Automation Interface, Element Package 2504
- Target
  - Role 967
  - Set For Message 959
- Types, Implementation Report 2342
- Task
  - Auto Naming 461
  - Auto Numbering 461
  - Automation Interface Repository 2471
  - Completion 450
  - Details 461
  - Element 1676
- Task Management
  - In Project Calendar 513
- TCF
  - Value 517
  - Weighting 517
- TcSE
  - MDG Integration For, Enterprise Architect 1289
- Team
  - Develop Projects In, Introduction 240
  - Development, Introduction 176
  - Development, Project Sharing 240
- Team Deployment
  - And Version Control 319
  - Version Control Branching 319
- Team Foundation Server
  - Version Control Option 352
- Team Review 285
  - Access 279
  - Access From Shortcut 183, 187
  - Access From Shortcut (Direct Definition) 185
  - Add External File Link To Post 292
  - Add New Category 284
  - Add New Post 287
  - Add New Topic 285
  - Add Object Link To Post 292
  - Add Team Review Link To Post 292
  - Capture Diagram Image As Resource 294
  - Connections To Other Team Reviews 296
  - Context Menu Options 280
  - Copy Path To Clipboard 280
  - Create Category 284
  - Create Post 287
  - Create Reply 289
  - Create Topic 285
  - Delete Category 280
  - Delete Item 291
  - Delete Post 280
  - Delete Resource 280
  - Delete Resources 294
  - Delete Topic 280
  - Edit Item 291
  - Editor 291
  - Export Package As Resource 294
  - Hyperlink To 1668

- Team Review 285
  - Icons 279
  - Import Resource Package 294
  - Introduction 279
  - Load Data When Required 296
  - Loading Behavior 296
  - Mark All Posts Unread 296
  - Options 296
  - Postings in Model Views 598
  - Preload 296
  - Reply To Post 289
  - Resources 294
  - Save Profiler Report As Resource 2175
  - Search 295
  - Tab 279
  - View Diagram Image Resource 294
  - Was Discussion Forum 279
  - Window 279
- Teamcenter Systems Engineering, Siemens PLM
  - MDG Integration For, Enterprise Architect 1289
- Technical Complexity Factor
  - Estimate Project Size 520
  - Value 517
  - Weighting 517
- Technology Developer
  - And Enterprise Architect 82
  - Project Role 82
- Technology Event
  - EA\_OnInitializeTechnologies 2663
- Technology Events
  - Add-In Model 2663
  - EA\_OnDeleteTechnology 2667
  - EA\_OnImportTechnology 2668
  - EA\_OnPostActivateTechnology 2665
  - EA\_OnPreActivateTechnology 2664
  - EA\_OnPreDeleteTechnology 2666
- Technology File
  - Generate (Option) 115
  - Import (Option) 115
- Technology-Defined Model View
  - Set Up 604
- Template
  - Behavioral Model 1449
  - Code Generation, Call From Other Template 1461
  - Editor In MDG Development 1462
  - Editor, Code Templates 1404
  - Element 1277
  - Fragments, HTML 2357
  - Instantiated 1183
  - Model, Incorporate In Technology 1355
  - Package, Settings 820
  - Parameterized Classes 1183
  - RTF, Import 2316
  - Script 2373
  - Transformation, Call From Other Template 1461
  - Transformation, Default 1713
- Template Binding
  - Connector 1257
- Term
  - Automation Interface Repository 2473
- Terminate
  - Element 1172
- Test
  - Application, Analyzer menu Option 120
  - Automation Interface, Element Package 2506
  - Copy Between Categories 2223
  - Create Defect From 2228
  - Documentation 2230
  - Field Substitution Macros 1438
  - Generate From Scenario 873
  - Import From Other Element 2226
  - Model Template 672
  - Move Between Categories 2223
  - Report 2230
  - Result Output 2230
  - Script Output 2230
  - Submenu (Analyzer) 124
  - Unit, In Execution Analysis 2183
  - Unit, Record Results In Execution Analysis 2185
  - Unit, Run In Execution Analysis 2185
  - Unit, Set Up In Execution Analysis 2183
- Test Case
  - Element 1677
- Test Cases
  - Generate From Scenario 873
- Test Cut 2187, 2191
  - Element 2198
  - Element, Create 2196
- Test Details Dialog 2214
- Test Domain 2187
  - Elements And Connectors 721
  - Group, Toolbox 721
- Test Domain Diagram
  - Create 2196
  - Generate 2196
- Test Script
  - Introduction 1834
  - JUnit 2183
  - NUnit 2183
- Test Scripts
  - Compartment 2229
- Test Set 2187



- Test Set 2187
  - Element 2198
  - Element, Create 2196
- Test Suite 2187
  - Element 2199
  - Element, Create 2196
- Tester
  - And Enterprise Architect 80
  - Project Role 80
- Testing
  - Acceptance 2220
  - Asterisk On Testing Window Tabs 2213
  - Autonaming 2213
  - Compartment, Element 842
  - Import Element Scenarios 2224
  - Import Internal Constraint 2227
  - Import Internal Requirement 2227
  - Import Package Scenarios 2224
  - Import Responsibility 2227
  - Integration 2216
  - MDG Technology For, Enterprise Architect 1289
  - Model Template 662, 672
  - Overview 2211
  - Report, Generate 2342
  - Scenario 2221
  - Support 2211
  - System 2218
  - Type, Define 1002
  - Unit 2215
  - Window 2213
  - Window, Acceptance Test Tab 2220
  - Window, Integration Test Tab 2216
  - Window, Scenario Test Tab 2221
  - Window, System Test Tab 2218
  - Window, Unit Test Tab 2215
  - Workspace 2213
- Testing Details Report
  - Generate 2343
- Testing Feature
  - Insert In Element 926
- Testpoint 1836
  - Combine 2196
  - Diagram 721
  - Introduction 2187
  - Management 2187
  - Toolbox Group 721
  - Window 2189
  - Window Toolbar 2191
- Testpoint Editor Dialog 2195
- Testpoint Management
  - Class Invariant Dialog 2193
  - Operation Constraints Dialog 2193
- Testpoint Trace
  - Create 2193
  - Level 2193
- Testpoints 1836
- Text Element
  - Create 814
  - Insert New From Toolbar 143
- Text File
  - Create Element From 1473
- TFS
  - Package Version Control History, In 376
  - Version Control Option 352
  - Working Folder For Enterprise Architect Version Control 352
  - Workspace For Enterprise Architect Version Control 352
- The Open Group Architecture Framework
  - MDG Technology For, Enterprise Architect 1289
- Theme
  - Microsoft Office Styles 162
  - Microsoft Visual Basic Styles 162
  - Options 162
  - Visual Style 98
- Tidy Line Angles 948
- Time Event 1149
- Time Interval
  - Compress 1069, 1071
  - Context Menu 1066
  - Copy and Paste 1071
  - Create 1067
  - Delete 1066
  - Description 1066
  - Move 1066
  - Operations 1071
  - Resize 1066
  - Select 1070
  - Shift Left Or Right 1071
  - Transitions 1071
- Time Range
  - Set For Timing Diagram 1053
- Timeline Element States
  - Add Via Configure Timeline Dialog 1062
  - Delete via Configure Timeline Dialog 1062
  - Edit Via Configure Timeline Dialog 1062
  - Maintain 1062
  - Numeric Range Generator 1064
- Timeline Range
  - Set For Timing Diagram 1053
- Timeline Start Position
  - Set For State Lifeline Element 1054

- Timing
  - Constraint 1238
  - Details, Change 1238
  - Elements and Connectors 711
  - Group, Toolbox 711
  - Message 1245
  - Message, Create 1246
  - Observation 1238
- Timing Diagram
  - Add Value Lifeline Element 1060
  - Create 1053
  - Description 1051
  - Edit Options 1054
  - Edit Value Lifeline Element 1060
  - Elements And Connectors 1051
  - Example 1052
  - Set Time Range 1053
- TOGAF
  - MDG Technology For, Enterprise Architect 1289
- Tomcat
  - Server, Configuration 1849
  - Server, Debugging 1846
  - Service Configuration 1850
- Toolbar
  - (File) Search, Debugging 1817
  - Add Commands 154
  - Add Menu Commands 155
  - Add Toolbar Commands 155
  - Change Command Icon Appearance 154
  - Code Generation 142
  - Console Tab, Scripting Window 2376
  - Create 155
  - Current Connector 148
  - Current Element 146
  - Customize 155
  - Debug & Record 150
  - Default Tools 139
  - Diagram 145
  - Diagram List 592
  - Display Labels 155
  - Docked 138
  - Element 143
  - Floating Buttons, Display 556
  - Floating, Structural Specification 864
  - Format (Element Appearance) 692
  - Hide 155
  - Large Icons 162
  - Manage Searches 621
  - Missing Icon 153
  - Model Search 611
  - Notes 979
  - Options, Customize Appearance 162
  - Package Browser 592
  - Project 141
  - Project Browser 583
  - Remove 155
  - Remove Commands 154
  - Rename 155
  - Screen Tips 162
  - Scripting Window 2373
  - Show 155
  - Submenu 98
  - Testpoints Window 2191
  - UML Elements 143
  - Workspace 138
  - Workspace Layouts 149
- Toolbar Customize Button
  - Show/Hide 98
- Toolbox
  - Activity Group 712
  - Add Stereotype To Diagram Element 698
  - Add To MDG Technology 1329
  - Analysis Group 715
  - Appearance Options 701
  - ArcGIS 1634
  - ArchiMate Group 1621
  - BPMN 1.0 1547
  - BPMN 1.1 1547
  - BPMN 2.0 1549
  - BPMN 2.0 Business Process 1550
  - BPMN 2.0 Choreography 1553
  - BPMN 2.0 Collaboration 1555
  - BPMN 2.0 Conversation 1557
  - BPMN 2.0 Type 1560
  - BPMN Group 1545
  - Business Modeling Group 1514
  - Class Group 708
  - Collapse Page 698
  - Common Group 706
  - Communication Group 710
  - Component Group 713
  - Composite Group 709
  - Connectors For Extending In Profile 1345
  - Create Elements And Connectors 698
  - Custom Group 716
  - Customize 1337
  - Data Flow Diagram Group 1506
  - Data Modeling Group 720
  - Default, Override In Profile 1342
  - Deployment Group 713
  - Documentation Group 2327
  - Elements For Extending 1344
  - Entity Relationship Diagram 1627

- Toolbox
  - Eriksson-Penker Group 1623
  - Expand Page 698
  - Gang Of Four Pattern Group 1928
  - GoF Pattern Group 1928
  - Hide 698
  - Hide Labels 701
  - ICONIX Group 1925
  - Interaction Group 710
  - Learning Center Profile, Define 1350
  - Maintenance Group 717
  - MDG Technology Groups 1284
  - Metamodel Group 715
  - Mind Mapping Group 1503
  - Object Group 709
  - ODM Pages 1644
  - Override Default In Toolbox Profile 1342
  - OWL Pages 1644
  - Page Attributes 1339
  - Page, Change Sequence In Profile 1337
  - Pin Pages 701
  - Profile Group 714
  - Profile, Change Icon Order 1337
  - Profile, Change Toolbox Page Order 1337
  - Profile, Create For MDG Technology 1337
  - Profiles 1337
  - RDF Pages 1644
  - Requirement Group 717
  - Set Toolbox Visibility 701
  - Shortcut Menu 704
  - Show 698
  - Show Labels 701
  - SoaML Pages 2088
  - SOMF Pages 2092
  - SPEM 524
  - State (Machine) Group 711
  - Synchronize Stereotyped Tagged Values 698
  - SysML Groups 1990
  - SystemC Group 1760
  - Test Domain Group 721
  - Timing Group 711
  - Unpin Pages 701
  - Use Case Group 707
  - User Interface Group 718
  - Verilog Group 1763
  - VHDL Group 1764
  - WSDL Group 719
  - XML Schema Group 720
- Toolbox Profile
  - Add Custom Compartments 1339
  - Connectors For Extending 1345
  - Create Hidden Submenu In 1341
  - Items, Assign Icons For 1342
  - Pages That Can be Overridden 1343
- Toolbqar
  - Record & Analyze Window 2163
- Tools
  - Custom 157
  - External Applications 157
  - Menu 115
  - Pass Parameters To External Tools 159
- Top Margin
  - Sequence Diagram, Change 1083
- Topic
  - Add To Team Review 285
  - Create 285
  - Delete 280
- TortoiseSVN
  - In Version Control 334
- Trace
  - Connector 1260
  - Relationship 1260
  - Statements 1873
- Trace Statements
  - Output 1860
- Trace Tools 566
- Traceability 636
  - Diagrams 649
  - For Diagram 637
  - For Elements 637
  - In Requirements Models 1494
  - Introduction 635
  - Themes 635
  - Tools 636
  - Window 637
  - Window, In Traceability 636
  - With Dependency Report 636
  - With Implementation Report 636
  - With Relationship Matrix 636
  - With Traceability Window 636
- Tracepoint
  - Statement Output 1860
- Tracepoints
  - In Code 1873
  - Set 1873
- Track Changes
  - Auditing 381
  - Baselines 381
  - Introduction 381
- Trademarks 31
- Transfer
  - Project Data Between Repositories 442
- Transform
  - Connector End 1727

- Transform
  - Connectors 1724
  - Copy Information 1727
  - Copy Tagged Values 1727
  - Duplication Of Connectors 1724
  - Elements, MDA-Style Transformations 1683
  - Model, MDA-Style Transformations 1683
  - Names 1728
  - Objects 1716
- TRANSFORM\_CLASSIFIER
  - Macro 1730
- TRANSFORM\_CURRENT
  - Macro 1724, 1727
- TRANSFORM\_REFERENCE
  - Macro 1724, 1730
- TRANSFORM\_TAGS
  - Macro 1727
- Transformation
  - Built In, MDA-Style Transformation 1686
  - C# 1687
  - Data Model To ERD 1689
  - DDL 1689
  - Debug Intermediary Language 1714
  - Dependencies 1679
  - EJB 1694
  - Entity Bean 1694
  - ERD To Data Model 1697
  - Java 1699
  - JUnit 1702
  - NUnit 1704
  - Session Bean 1694
  - Template, Call From Other Template 1461
  - Write 1712
  - WSDL 1705
  - XSD 1707
- Transformation Dependency
  - Trace With Traceability Window 637
- Transformation Template
  - Default 1713
  - Modify 1710
  - Transfer Between Models, Export 1400
  - Transfer Between Models, Import 1401
- Transition
  - Add To State Lifeline Elements 1057
  - Add Via Configure Timeline Dialog 1065
  - Change In State Machine Table 1045
  - Change Time, State Lifeline Element 1057
  - Connector 1260
  - Context Menu Options 938
  - Delete On State Lifeline Element 1057
  - Delete Via Configure Timeline Dialog 1065
  - Edit In Time Intervals 1071
  - Edit On State Lifeline Elements 1057
  - Edit Via Configure Timeline Dialog 1065
  - Effect 1260
  - Guard 1260
  - Highlight Associated Trigger or State 1045
  - Insert In State Machine Table 1045
  - Locate In State Machine Diagram 1047
  - Locate In State Machine Table 1047
  - Merge On State Lifeline Element 1057
  - Move On State Lifeline Elements 1057
  - Properties 1260
  - Relationship 1260
  - State Machine Table Conventions 1047
  - Trigger 1260
- Transitions Collection
  - Automation Interface, ElementFeatures Package 2412
- Translation
  - RTF Report Generation 2269
- Tree Style Hierarchy
  - Create 960
  - Set Default Link Style 960
- Trial Version
  - Extend Trial Period 20
  - Of Enterprise Architect 20
  - Select Edition To Trial 20
  - Select Workspace Layout 20
- Triangle
  - Red 530
- Tricks and Traps
  - Create Add-In 2591
- Trigger
  - Create 1173
  - Create In State Machine Table 1044
  - Create In Transition Properties 1260
  - Data Modeling, Create 1965
  - Data Modeling, What Is A? 1965
  - Element 1173
  - Find In Project Browser 1260
  - For Transition 1260
  - Intermediate Event, BPEL 1.1 1578
  - Intermediate Event, BPEL 2.0 1602
  - Locate 2122
  - Locate In State Machine Diagram 1047
  - Locate In State Machine Table 1047
  - Ports 1173
  - Properties Tab 1173
  - Reposition In State Machine Table 1046
  - Simulation 2122
  - Simulation Events Window 2122
  - Start Event, BPEL 1.1 1577
  - Start Event, BPEL 2.0 1601

- Trigger
    - State Machine Table Conventions 1047
    - Type 1260
    - Use Of 2122
  - Trigger Property
    - Find In Project Browser 938
  - Type Hierarchy Dialog 801
  - Type Libraries
    - For Script Editor 2378
  - Type Specific Menu Section
    - Connector 938
- U -
- UAC
    - And Debugging 1838
  - UI Control
    - Element 1657
  - Ultimate Edition
    - Of Enterprise Architect 22
  - UML
    - 1.3 411, 413, 535
    - 1.3, Import From XMI 417
    - 1.4 413
    - 1.4, Import From XMI 417
    - 1.5 556
    - 2.0 Migration 535
    - 2.0, Import From XMI 417
    - 2.3 - Definition 9
    - Analysis Tool - Enterprise Architect 5
    - Build Systems 9
    - Connectors 1206
    - Data Modeling Profile 1625
    - Definition 1008
    - Design Systems 9
    - Design Tool - Enterprise Architect 5
    - Dictionary 1008
    - DTD 422
    - Elements 1090
    - Enterprise Architect Modeling Platform 657
    - Extend 1008
    - Manage Complexity 9
    - Mappings To XSD 2048
    - Model Complexity 9
    - Model Structure, Manage - Enterprise Architect 9
    - Models Under Single Root 419
    - Recommended Reading 1008
    - Shared Model Development - Enterprise Architect 9
    - Support - Enterprise Architect 9
    - Syntax Checking Option 549
    - Visualize Systems - Enterprise Architect 9
  - UML 2.1
    - Data Modeling Notation 1982
  - UML Behavioral Diagram
    - Overview 1027
  - UML Business Process Model
    - In Enterprise Architect 662, 665
  - UML Class Model
    - In Enterprise Architect 662, 668
  - UML Component Model
    - In Enterprise Architect 662, 670
  - UML Database Model
    - In Enterprise Architect 662, 669
  - UML Deployment Model
    - In Enterprise Architect 662, 671
  - UML Diagram
    - Add To Project 724
    - Copy, Deep 742
    - Copy, Shallow 742
    - Create 724
    - Duplicate 742
    - Extended 1010, 1502
    - In Enterprise Architect 687
    - MDG Technology 1010
    - Overview 1010
    - Paste 742
    - Types 1010
    - What Is A? 1010
  - UML Domain Model
    - In Enterprise Architect 662, 667
  - UML Element
    - Behavioral Diagram Elements 1090
    - Structural Diagram Elements 1179
    - Toolbar 143
  - UML Maintenance Model
    - In Enterprise Architect 662, 673
  - UML Model
    - Add Connectors In Enterprise Architect, Tutorial 55
    - Add Diagram In Enterprise Architect, Tutorial 52
    - Add Element In Enterprise Architect, Tutorial 53
    - Add Element To Diagram, Tutorial 53
    - Add Package In Enterprise Architect, Tutorial 52
    - Add View, Tutorial 51
    - CSV Import And Export In Enterprise Architect 435
    - Define Connector Properties In Enterprise Architect 56
    - Define Element Properties In Enterprise Architect 56

- UML Model
  - Domain Based 1498
  - Project Roles And Tasks In Enterprise Architect 72
  - Specialized 1498
- UML Model Pattern
  - Introduction 662
- UML Model Template
  - In Enterprise Architect 662
  - Introduction 662
- UML Modeling
  - And MDG Technologies 657
  - And Requirements Management In Enterprise Architect 657
  - Build Models 657
  - Business Modeling In Enterprise Architect 657
  - What Is? 657
  - With Enterprise Architect 657
  - With Packages 680
  - With UML Patterns In Enterprise Architect 657
  - With UML Profiles In Enterprise Architect 657
  - With UML Stereotypes In Enterprise Architect 657
- UML Modeling Tool
  - Enterprise Architect - Key Features 14
- UML Pattern
  - Actions 1271
  - Add To Diagram 1274, 1276
  - Create From Diagram 1271
  - In Resources View 1271
  - Save 1271
  - Save From Diagram 1271
- UML Profile
  - And Element Templates 1277, 1293
  - Import 1279
  - Import From XML 1277, 1293
  - Stereotypes 1277, 1293
  - Synchronize Elements 1281
  - Synchronize Stereotypes 1281
  - Synchronize Tagged Values And Constraints 1281
  - XSD, Stereotypes 2055
- UML Project Management Model
  - In Enterprise Architect 662
- UML Project Model
  - In Enterprise Architect 674
- UML Requirements Model
  - In Enterprise Architect 662, 666
- UML Resources
  - Patterns 1271, 1273
- UML Structural Diagram
  - Overview 1011
- UML Syntax Compliance
  - Turn Off 2205
- UML Testing Model
  - In Enterprise Architect 662, 672
- UML Types
  - Cardinality (Multiplicity) 986
  - Dialog 981
  - Stereotypes 982
  - Tagged Values 985
- UML Use Case Model
  - In Enterprise Architect 662, 666
- UML\_EA.DTD 422
  - File 417
- UML2
  - File Import 419
- Unadjusted Use Case Points 520
- Undo
  - Last Action, Diagram Edits 768
  - Option (Edit Menu) 93
- Unicode Support 47, 544
- Unified Modeling Language 1008
- Unit Test Command
  - Introduction 1834
- Unit Test Script
  - Create 1834
- Unit Testing 2215
  - Define Tests, Execution Analysis 2183
  - Introduction, Execution Analysis 2183
  - JUnit 2183
  - NUnit 2183
  - Record Test Results, Execution Analysis 2185
  - Run, Execution Analysis 2185
  - Set Up, Execution Analysis 2183
- Unlock
  - Connector 268
  - Element 268
- Unpin
  - Toolbox Pages 701
- Update
  - Element Phase, For Package 529
  - Element Status, For Package 529
  - Element Version, For Package 529
  - Package Phase 529
  - Package Status 529
  - Package Version 529
- Upgrade
  - Existing License (v. 6.5 And Earlier) 2743
  - Existing License (v. 7.0+) 2743
  - Model 538
  - Model, Upgrade Wizard 537
  - Project 537
  - Replicas 249, 538

- Upgrade
  - Wizard 537
- Upsize
  - From Desktop and Professional Editions 193
  - To Access 2007 227
  - To MySQL 224
  - To Oracle 230
  - To PostgreSQL 232
  - To Progress OpenEdge 236
  - To SQL Server 228
  - To SQL Server Desktop Engine (MSDE) 236
  - To Sybase Adaptive Server Anywhere (ASA) 234
- Usage
  - Connector 1263
  - Of Element 802
  - Relationship 1263
- Use
  - Classifiers 899
  - Connector 1263
  - Pattern 1274
  - Relationship 1263
  - Spell Checker 485
- Use Case
  - And Requirements 1483
  - Arrowhead, Show 959
  - Create For Method 1807
  - Element 1174
  - Elements and Connectors 707
  - Extension Points 1176
  - Group, Toolbox 707
  - Keyword 520
  - Link To Method 1807
  - Metrics 516
  - Metrics Dialog 520
  - Model Template 662, 666
  - Phase 520
  - Points, Unadjusted 520
  - Quick Generation Of Realization 1480
  - Scenarios 1078
  - View 676
- Use Case Diagram
  - Description 1030
  - Elements And Connectors 1030
  - Example 1032
- Use Element Extras
  - Automation Interface Code Example 2575
- Use Repository Extras
  - Automation Interface Code Example 2578
- User
  - Default Fonts 552
  - Defined Searches, Storage 623
  - Dictionary, For Spell Checker 486
  - Directory 989
  - Forum 42
  - Groups 256
  - ID, Import From Active Directory 257
  - Settings 1901
- User Interface
  - Components 86
  - Control Element 1657
  - Element 1657
  - Elements and Connectors 718
  - Group, Toolbox 718
  - MDG Win32® User Interface Technology 1661
  - Model Template 662
  - Screen Prototype 1656
- User Interface Diagram
  - Description 1655
  - Elements And Connectors 1655
  - Example 1656
- User Lock
  - Identify Owner 277
  - Indicators 276
- User Security
  - Add Connector To Locked Element 273
  - Apply User Lock 274
  - Assign User To Group 260
  - Basics 251
  - Change Password 270
  - Disable 253
  - Enable 253
  - Identify Lock Owner 277
  - Import User ID From Active Directory 257
  - List Of Permissions 264
  - Lock Elements & Diagrams 271
  - Lock Packages 273
  - Locked Element Indicators 276
  - Maintain Groups 262
  - Maintain Users 256
  - Manage User-level Locks. 278
  - Password Encryption 269
  - Policy 255
  - Re-enable 253
  - Release Elements & Diagrams 271
  - Reset Password 270
  - Set Group Permissions 263
  - Set Password 270
  - Set Up Single Permissions 260
  - Tasks 251
  - View All Permissions 261
  - View And Manage Locks 268
  - What Is User Security? 251
- User Security Groups

- User Security Groups
    - Assign User To 260
    - Set Up 260
  - User Settings
    - Options Dialog 153
  - User/Group Lock
    - Release 271
    - Set 271
  - User/Group Lock Policy 255
  - Uses Context List Facility 861
  - Using Enterprise Architect
    - Application Workspace 86
    - Remove Recent Project 89
    - Start Page 87
  - Utility
    - Compare 394
    - Diff 394
  - UUCP
    - Estimate Project Size 520
- V -**
- Validate
    - BPEL Model 1619
    - Business Rules 1542
    - Model To Generate BPEL 1.1 1594
    - Package Configuration For Version Control 379
    - Rule Composer 1542
  - Validation
    - Cancel 102
    - Of Diagram 2203
    - Of Element 2203
    - Of Model, Configure 2205
    - Of Model, Configure For MDG Technology 1354
    - Of Package 2203
    - Rules, Model 2206
  - Validation, Properties
    - Attribute 2208
    - Element 2208
    - Feature 2208
    - Relationship 2208
  - Validation, Well Formedness
    - Attribute 2207
    - Diagram 2207
    - Element 2207
    - Feature 2207
    - Relationship 2207
  - Value Lifeline Element 1177
    - Add States 1061
    - Add To Timing Diagram 1060
    - Add Transitions 1061
    - Change Transition Time 1061
    - Delete Transitions 1061
    - Edit Transitions 1061
    - Sizing and Scale 1060
    - States 1060
    - Transitions 1060
  - ValueOf Field
    - RTF Style Template Editor 2314
  - Variable
    - Comparison Of Values 1881
    - Debug, Break On Change In Value 1882
    - Definitions, Code Template Syntax 1407
    - Definitions, Examples 1407
    - References, Code Template Syntax 1407
    - References, Examples 1407
    - Snapshots 1881
  - VB
    - Set Up In Automation Interface 2385
  - VB.NET
    - Code Generation 1918
    - Language Options 1918
    - Modeling Conventions 1761
  - VBScript 2372, 2373
  - Verilog
    - Code Generation 1919
    - Enterprise Architect Toolbox Pages 1763
    - Language Options 1919
    - Modeling Conventions 1763
  - Version Control
    - And Reference Data 317
    - And Replication 353
    - And Sequence Diagram 1079
    - Apply To Enterprise Architect Model 316
    - Apply To Model Branch 371
    - Basics 318
    - Branching 319
    - Configuration 325, 344, 346, 372
    - Configuration, Team Deployment 319
    - Configuration, Use Previously-Defined 346
    - Configure Current Package 355
    - Configure Package 362
    - Considerations 313
    - Controlled Packages 422
    - Copy-Modify-Merge Policy 318
    - CVS Options 350
    - Discussion Of File Control 316
    - Export Model Branch 371
    - Features 312
    - File History 357
    - File Properties 357
    - Import Model Branch 372
    - In Team Deployment 319



- Version Control
  - Include Other Users' Packages 370
  - Introduction 313
  - Locking - Necessary? 318
  - Lock-Modify-Unlock Policy 318
  - Manual, With XML 432
  - Menu (Package) 357
  - Menu (Project) 355
  - Nested Packages 316
  - Offline 322
  - Package Configuration, Validate 379
  - Policies 318
  - Project Browser Indicators 321
  - Recommendations 353
  - Refresh View Of Shared Model 242
  - Resynchronize Package Version Control Status 380
  - Resynchronize Status Of All Packages 355
  - Review Version Control History (SCC) 377
  - Review Version Control History (TFS, CVS, Subversion) 376
  - Revision, Check Out 376
  - SCC Options 348
  - SCC, Providers Dialog SCC, Providers Dialog 348
  - SCC, Upgrade For Enterprise Architect 4.5 344
  - Set Up 344
  - Settings 346, 355
  - Setup Menu 355
  - Submenu (Project Menu) 103
  - Subversion Options 351
  - Subversion, Create Local Working Copy 330
  - Subversion, Create New Repository Subtree 329
  - Subversion, TortoiseSVN 334
  - Subversion, Using With Enterprise Architect Under WINE Crossover 332
  - System Requirements 325
  - Tasks 353
  - TFS Option 352
  - Usage Scenarios 315
  - Use Nested Version Control Packages 320
  - Using 353
  - Validate Package Configurations 355
  - Who Has Checked Out A Package? 357
  - Work Offline 355
- Version Controlled Package
  - Icon 585
- VHDL
  - Code Generation 1919
  - Enterprise Architect Toolbox Pages 1764
  - Language Options 1919
- Modeling Conventions 1764
- View
  - Add To UML Model, Quick Start 51
  - Audit 387
  - Data Modeling, What Is A? 1967
  - Diagram List 596
  - Locks 268
  - Menu 95
  - Next Diagram 764
  - Of Model 598
  - Package Browser 587
  - Previous Diagram 764
  - Project Statistics 100
  - Requirements 1489
  - Submenus 98
  - Tabs, Close 695
- View Elements Of Array
  - Debugger 1881
- View Header
  - Model Search 619
- View Options
  - Diagram List 596
  - Diagram View 691
  - Package Browser 587
- View Tab Switcher 697
- Views
  - Add 677
  - Class 676
  - Component 676
  - Delete 678
  - Deployment 676
  - Dynamic 676
  - Manage 676, 677
  - Rename 678
  - Simple 676
  - Use Case 676
- Virtual Document
  - Add Packages As Attributes 2333
  - Change Package Sequence 2335
  - Consecutive Numbering Through 2337
  - Create Master Document For 2330
  - Create Model Document For 2331
  - Delete Package Attributes 2334
  - Delete Packages 2334
  - Document Order 2335
  - Generate HTML Report From 2368
  - Generate RTF Report From 2338
  - Introduction 2327
  - Master Document Element 2327
  - Model Document Element 2327
  - Model Search Sequence 2335

- Virtual Document
  - Move Package Attributes Between Elements 2335
  - Numbering Through 2337
  - RTF Bookmarking In 2327
  - Sequence of Model Document Elements 2335
  - Sequence of Package Attributes 2335
  - Tagged Values For Master Document 2330
- Virtual Document (HTML) 2368
- Virtual Machine 1843
- Visibility Indicators 731
  - Values 1741
- Visibility Of Elements
  - Customize 813
- Visible Class Members
  - Set Diagram Appearance Options 732
- Visio
  - MDG Link For, Enterprise Architect 1289
- Vista
  - Permissions For Enterprise Architect 36
- Visual Basic
  - Code Generation 1920
  - Connect To Automation Interface 2382
  - Debugging, General Setup 1841
  - Import, Reverse Engineering 1793
  - Language Options 1920
  - Modeling Conventions 1766
  - Set Up In Automation Interface 2385
- Visual Basic.Net
  - Import, Reverse Engineering 1793
- Visual Check
  - Diagram Against Baseline 403
- Visual Execution Analyzer
  - Access 2143
  - Advanced Debug Techniques, Java 1845
  - Availability 2143
  - Break On Variable Changing Value 1882
  - Breakpoint Management 1869
  - Breakpoints And Markers Window, Record Sequence Diagrams 1869, 2157
  - Build Script, Create 1832
  - Control Recording, Record Sequence Diagrams 2161
  - Create Sequence Diagram, Call Stack 1886
  - Debug .NET 1850
  - Debug .NET CLR Versions 1851
  - Debug .NET With COM Interop Process 1852
  - Debug Another Process 1888
  - Debug Apache Tomcat Server Configuration 1849
  - Debug Apache Tomcat Windows Service 1850
  - Debug ASP .NET 1853
  - Debug Java 1843
  - Debug Java Applets In Internet Browsers 1845
  - Debug Java Web Servers 1846
  - Debug JBOSS Server Configuration 1848
  - Debug Symbols, C++ And Native Applications 1842
  - Debugger Frameworks 1867
  - Debugger System Requirements 1837
  - Debugger, Overview 1867
  - Debugging Actions 1874
  - Deploy Script, Create New 1862
  - Diagram Features, Generate Sequence Diagrams 2152
  - Difference Between Recording Marker And Breakpoint 2155
  - File Search, Use 1817
    - For C++ Applications 1841
    - For Microsoft Native Applications 1841
    - For WINE Applications 1839
  - General Debug Setup, Java 1843
  - Generate Sequence Diagram 2164
  - Inspect Process Memory 1886
  - Introduction 2143
  - Java Debug Session, Attach To VM 1845
  - Map State Changes 2168
  - Marker Management 1869
  - Marker Types, Record Sequence Diagrams 2155
    - MDDE Basic Setup 1827
    - MDDE External Tools 1827
    - MDDE, Analyzer Scripts, Manage 1828
    - MDDE, Build Scripts 1832
    - MDDE, Generate Code 1825
    - MDDE, Script Actions, Define 1831
    - MDDE, Synchronize Code 1825
  - Model Simulation 2100
  - Object Workbench, Create Variables 2177
  - Object Workbench, Introduction 2176
  - Object Workbench, Invoke Methods 2179
  - Object Workbench, Overview 2176
  - Outputs 2143
  - Overview 2143
  - Place Markers, Recording Sequence Diagrams 2154
  - Profiler Overview 2170
  - Record Activity For Class 2160
  - Record Activity For Method 2155
  - Record Sequence Diagrams, Introduction 2147
  - Record Sequence Diagrams, Overview 2148
  - Record Sequence Diagrams, Recording Options 1863
  - Record Sequence Diagrams, Set Up 2153

- Visual Execution Analyzer
  - Record State Changes 2168
  - Record State Transitions 2165
  - Record Unit Test Results 2185
  - Recording History 2149
  - Recording Markers, Activate, Record Sequence Diagrams 1869
  - Recording Markers, Disable, Record Sequence Diagrams 1869
  - Recursive Builds 1832
  - Run Script, Create New 1862
  - Run Unit Test 2185
  - Sample Model, Generate 2145
  - Save Recording History 2164
  - Script Search 1817
  - Search Window 1817
  - Sequence Diagrams, Limit Auto Recording 2155
  - Set Code Breakpoint 1871
  - Set Data Breakpoint 1882
  - Set Up Debug Session 1837
  - Set Up Debug Session For .NET 1850
  - Set Up To Capture State Changes 2166
  - Show Loaded Modules 1887
  - Show Output 1875
  - Start Debugger 1875
  - State Machine 2166
  - State Transitions 2166
  - Step Into Function Calls 1875
  - Step Out Of Functions 1875
  - Step Over Lines Of Code 1875
  - Step Though Function Calls 2162
  - Stop Debugger 1875
  - Tooltips In Code Editor 1879
  - UAC-Enabled Operating Systems 1838
  - Unit Test Script, Create 1834
  - Unit Test Script, Introduction 1834
  - Unit Test Script, Set Up 2183
  - Unit Test, Record Results 2185
  - Unit Testing, Introduction 2183
  - Uses Of 2143
  - View Call Stack 1884
  - View Local Variables 1877
  - View Local Variables, Long Values 1879
  - View Variables In Other Scopes 1880
  - Work With Marker Sets, Record Sequence Diagrams 2158
  - Workspace Layouts 1826
- Visual Execution Profiler
  - Attach To Process 2173
  - Getting Started 2171
  - Launch 2173
  - Operation 2174
  - Overview 2170
  - Prerequisites 2171
  - Report, Example 2170
  - Report, Load 2175
  - Report, Save 2175
  - Report, Save As Resource In Team Review 2175
  - Set Options 2174
  - Set Sample Intervals 2174
  - Start 2173
  - Stop 2173
  - Supported Platforms 2171
  - System Requirements 2171
  - Team Review, Save Report As Resource 2175
  - Toolbar 2171
- Visual Execution Sampler
  - Attach To Process 2173
  - Getting Started 2171
  - Launch 2173
  - Operation 2174
  - Overview 2170
  - Prerequisites 2171
  - Report, Example 2170
  - Report, Load 2175
  - Report, Save 2175
  - Report, Save As Resource In Team Review 2175
  - Set Options 2174
  - Set Sample Intervals 2174
  - Start 2173
  - Stop 2173
  - Supported Platforms 2171
  - System Requirements 2171
  - Team Review, Save Report As Resource 2175
  - Toolbar 2171
- Visual Representation
  - Of Elements, Introduction 841
- Visual Studio 2005/2008
  - MDG Integration For, Enterprise Architect 1289
- Visual Studio.NET
  - MDG Link For, Enterprise Architect 1289
- Visual Styles
  - Microsoft Office Styles 162
  - Microsoft Visual Basic Styles 162
  - Options 162
  - Select 162
  - Submenu 98
  - Themes 98
  - View Menu Option 95
- Visualize Systems
  - Enterprise Architect 9

- VM
  - Attach To In Java Debug Session 1845
  
- W -**
  
- W3C XSD
  - Elements and Connectors, Enterprise Architect 720
- W3CWSDL
  - Elements And Connectors, Enterprise Architect 719
- WAN Optimization
  - Enterprise Architect Performance 210, 212, 214, 216, 218, 219, 221
  - For ASA Data Repository 219
  - For MSDE Server Data Repository 221
  - For MySQL Data Repository 210
  - For Oracle Data Repository 214, 216
  - For PostgreSQL Data Repository 218
  - For Progress OpenEdge Data Repository 221
  - For SQL Server Data Repository 212
- WAN Optimizer
  - Introduction 238
  - Performance Of Enterprise Architect 238
  - Transmission Diagram 238
- Watched Items
  - Debugger 1880
- Watches Window 1880
  - Break On Variable Changing Value 1882
- Watermark
  - Diagrams 550
- Web
  - Page Modeling 1660
  - Report Style Templates 2355
  - Stereotypes 1660
  - Templates, Reporting 2355
- Web Browser
  - Home Website, Access 165
  - Internet Email, Access 165
  - Web Search Engine, Access 165
- Web Ontology Language 1643
  - Definition Diagram, Example 1651
  - Facts Diagram, Example 1651
- Web Page
  - Prototype 1656
- Web Server
  - Java, Debug 1846
- Web Service
  - Create In BPEL 1.1 1591
  - Submenu (Tools Menu) 118
- Web Service Definition Language 2067
- Web Service Operation
  - BPEL 2.0, Create 1615
- Web Services (WSDL) 2067
  - Generate WSDL 2085
  - Import WSDL Files 2086
  - Model WSDL 2069
  - Model WSDL, Binding 2078
  - Model WSDL, Document 2083
  - Model WSDL, Message 2073
  - Model WSDL, Message Part 2074
  - Model WSDL, Namepaces 2067, 2070
  - Model WSDL, Port Type 2075
  - Model WSDL, Port Type Operation 2076
  - Model WSDL, Service 2082
- Web Services Business Process Execution Language (WS-BPEL) 1570
- Web Site
  - Access Any 165
  - Access Home 165
  - Home, Define Default 544
  - Sparx Systems 40
- website SOAML document page. 2088
- Well Formedness Validation
  - Attribute 2207
  - Diagram 2207
  - Element 2207
  - Feature 2207
  - Relationship 2207
- What Is
  - A Connector? 1206
  - A Foreign Key? 1956
  - A Model? 660
  - A Pattern? 1271
  - A Primary Key? 1952
  - A Project? 179
  - A Requirement? 1468
  - A Stored Procedure? 1961
  - A Tagged Value 969
  - Enterprise Architect? 5
  - User Security? 251
  - XMI? 411
- Whiteboard
  - Diagram Mode 727
  - Diagram Mode, API 1348
- Whitespace
  - In Package Names 1777
- Wide Area Network (WAN) Optimizer
  - Introduction 238
  - Performance Of Enterprise Architect 238
  - Transmission Diagram 238
- Win32® User Interface
  - MDG Technology 1661
- Window

- Window
  - Autohide 137
  - Breakpoints And Markers 1869
  - Breakpoints And Markers, Record Sequence Diagrams 2157
  - Bring To Top 129
  - Call Stack 1879, 1884
  - Close All 129
  - Close All Except Current 129
  - Close Current 129
  - Combine In Frame 135
  - Connections 648
  - Constraints 879
  - Debug, Step Over Lines 1875
  - Diagram Filter 631
  - Dock 135
  - Element Browser 876
  - Floating 135
  - Generalized Functions 164
  - Hide 129
  - Layout Tools 769
  - Learning Center 89
  - Links 648
  - Locals 1877
  - Locals, View Long Values 1879
  - Maintenance 2237
  - Make Active 129
  - Memory Viewer 1886
  - Menu 129
  - Model Views 598
  - Modules 1887
  - Notes 977
  - Output 164
  - Output, DebuggerCode 1875
  - Pan And Zoom 610
  - Personal Information 486
  - Project Management 448
  - Project Task Allocation 472
  - Properties (Element) 878
  - Record & Analyze 2149
  - Relationships 648
  - Reload 129
  - Remove From Frame 135
  - Requirements 879
  - Resources 1003
  - Reveal Autohidden 137
  - Scenarios 879
  - Scenarios & Requirements 879
  - Scripting 2372
  - Search, Debugging 1817
  - System 460
  - Tagged Value 969
  - Team Review 279
  - Testpoints 2189
  - Traceability 637
  - Watches 1880
  - Watches, Break On Variable Changing Value 1882
- Windows
  - Authentication 256
  - Service, Apache Tomcat 1850
- Windows 7
  - Use Debugger 1838
- Windows Active Directory
  - Import User Login ID From 257
- Windows Authentication
  - Accept 257
- Windows Vista
  - Permissions For Enterprise Architect 36
  - Use Debugger 1838
- WINE
  - Debugging 1839
  - DIB Data Access Violation 1839
- WINE-Crossover
  - Using Subversion With Enterprise Architect Under 332
- Word
  - Special Considerations 2347
- Word Substitution
  - RTF Report Generation 2269
- Work Flow
  - Monitor With Model Views 532
- Work With
  - Attributes, Automation Interface Code Example 2581
  - Enterprise Architect 7
  - Methods, Automation Interface Code Example 2582
- workbench 1861
- Workbench Variables
  - Constructors 2177
  - Create 2177
- Workflow
  - Data Structures 299
  - Monitor 498
  - Objects 299
  - Script Functions 299
  - Script, Import 2373
  - Scripts 2375
  - Scripts, Execute 498
  - Scripts, Introduction 298
  - Searches 303
  - Status 303
  - Tab, Personal Information Window 498

- Workflow Script
    - Search Functions With User Tasks 301
    - System-Filled Data Structures 301
    - User Filled Data Structures 303
    - User Input Validate And Control Functions 300
    - User-Called Functions 304
  - Working
    - With MDG Technologies 1284
    - With UML Connectors 936
  - Working Sets
    - Create 498
    - Delete 498
    - History 498
    - Open 498
    - Overview 498
    - Tab, Personal Information Window 498
  - Workspace
    - Toolbars, Docked 138
    - Toolbars, Introduction 138
  - Workspace Layout
    - For Execution Analysis 1826
  - Workspace Layouts
    - Apply 149
    - Capture 149
    - Customize 149
    - Delete 149
    - Manage 149
    - Select 149
    - Show/Hide From Toolbar 149
    - System Provided 149
    - Toolbar 149
    - Upgrade Customized 149
    - User Defined 149
    - View Menu Option 95
  - WS-BPEL 1570
  - WSDL
    - Binding Diagram 2078
    - Binding Element 2078
    - Document Element 2083
    - Elements And Connectors 719
    - Group, Toolbox 719
    - Import 2085, 2086
    - Message Diagram 2073
    - Message Element 2073
    - Model 2069
    - Namespace Element 2067, 2070
    - Overview Diagram 2070
    - Port Type Diagram 2075
    - Port Type Element 2075
    - Port Type Operation 2076
    - Service Diagram 2082
    - Service Element 2082
    - Split Files 2086
    - Toolbox Pages 2069
    - Transformation 1705
    - Web Services 2069
    - WSDL Support
      - Introduction 2067
- X -**
- XMI
    - Export 411, 413
    - Export MOF To 2097
    - Export To 414
    - Import 411, 417
    - Import And Auditing 393
    - Import MOF From 2097
    - Limitations 421
    - Manual Version Control 432
    - Report Deletion Of Cross Package References 433
    - Specifications 411
    - UML DTD 422
    - XMI 1.1 413
    - XMI 2.1 413
  - XMI Import
    - Impact On Auto Increment Columns 536
  - XMIType Enum
    - Automation Interface 2423
  - XML
    - BPMN 2.0 1569
    - Code Page 562
    - Default XMI Version 562
    - Documents, Default Editor 562
    - Export To 414
    - Import ArcGIS Workspace 1641
    - Import Referenced Schema 2053
    - Package 422
    - Pattern File 1271
    - Set Default XML Directory 562
    - Set Default XML Editor 562
    - Specifications, Options Dialog 562
    - Structure Tree In Source Code Viewer 1799
    - XMI 1.0 Prefix 562
  - XML Engineering 2026
  - XML Schema
    - Elements and Connectors 720
    - Generate In Garden Of Eden Style 2051
    - Group, Toolbox 720
    - Submenu (Tools Menu) 118
    - UML Profile For XSD 2055
    - XSD 2027
  - XSD

## XSD

- Abstract Models 2046
  - Datatype Packages 2065
  - Generate 2049
  - Import 2053
  - Import, Global Element Behaviour 2054
  - Model 2027
  - Transformation 1707
  - XML Schema 2027
- XSDany 2055
- XSDattribute 2055
- XSDattributeGroup 2055
- XSDchoice 2055
- XSDcomplexType 2055
- XSDelement 2055
- XSDgroup 2055
- XSDrestriction 2055
- XSDschema 2055
- XSDsequence 2055
- XSDsimpleType 2055
- XSDtopLevelAttribute 2055
- XSDtopLevelElement 2055
- XSDunion 2055

**- Z -**

## Zachman

- Profile 753

## Zachman Framework

- MDG Technology For, Enterprise Architect 1289

## Zoom

- Control On Status Bar 152
- Diagram From Toolbar 145
- Diagram View 763
- Submenu 763

## Zooming

- Code Editor, Common 1810

## Z-Order

- Diagram Structure 687
- Element 744
- Modify, Of Elements 687
- Submenu 112





# Enterprise Architect User Guide

[www.sparxsystems.com](http://www.sparxsystems.com)