



ENTERPRISE ARCHITECT

User Guide Series

Model Exchange

Author: Sparx Systems

Date: 2026-05-28

Version: 17.2

CREATED WITH  **ENTERPRISE
ARCHITECT**

Table of Contents

Model Exchange	3
Model Exchange Quick Start	4
Model Exchange in XMI Format	6
Export to XMI	8
Import from XMI	10
Merge with XMI Using Merge File	13
Merge with XMI	15
Import EMX/UML2 Files	17
Import a Rhapsody Model	19
Import a MagicDraw MDZIP File	20
Limitations of XMI	22
The UML DTD	23
Model Exchange with Custom Package Sets	24
Transfer a Complete Repository	26
Transfer a Repository in Native Format	28
Native File Formats	31
Migrate an EAP/EAPX File to QEA File Format	35
Project Upgrade to QEA	37
Transfer a QEA/QEAX Repository to FEAP Format	38
Model Publishing to Various Formats	40
Using Controlled Packages	43
Controlled Package Menu	44
Configure Packages	45
Remove Package from Control	47
Save a Package	48
Load a Package	49
Batch Export Controlled Packages	50
Batch Import Controlled Packages	51
Manual Version Control with XMI	52
Report Deletion of Cross Package References	54
CSV Import and Export	56
CSV Specifications	57
Using Preserve Hierarchy	60
CSV Export	61
CSV Import	63

Model Exchange

Using Enterprise Architect, you can transfer data between projects using many different methods:

- **XML Import and Export** - Packages can be exported from and imported into your models using standard or Enterprise Architect specific XML formats; this greatly improves the flexibility and robustness of the models because Analysts and Modelers can externalize model elements in XML format for Version Control, distributed development, post processing and transferring Packages between models
- **Generate an XML file from a Model Document element** to enable a custom set of Packages to be imported into another model together rather than having to generate each one individually
- **Repository Data Transfer** - You can move complete models between files, and between files and repositories, row by row, table by table
- **CSV Import and Export** - You can import and export information on Enterprise Architect elements in CSV format, and then store and retrieve the information from spreadsheet tools such as Microsoft Excel
- **Using the Data Miner** - You can abstract data from a range of external data sources including databases (ODBC, ADO, OLEDB, JET), text files (XML, JSON, plain text), Excel (xls, CSV), and online files or URLs, creating a reproducible import process defined in a model format

Model Exchange Quick Start

It is possible to copy and move individual Packages between projects by exporting and importing the Packages as Native/XMI files. You can also duplicate larger structures, such as complete models or 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

Ribbon	Publish > Model Exchange > Export > Native Format Publish > Model Exchange > Export > XMI Format Publish > Model Exchange > Import > Native File Publish > Model Exchange > Import > XMI File
Keyboard Shortcuts	Ctrl+Alt+E (Export to Native/XMI Format) Ctrl+Alt+I (Import Native/XMI File)

Copy a Package from one Enterprise Architect Project to Another

Step	Action
1	Open the project to copy from.
2	In the Browser window, click on the Package to copy and use your preferred Export access method to open the 'Export Package to Native/XMI File' dialog.
3	Select the appropriate options and filename.
4	Click on the Export button to begin the export process.
5	When the export is complete, open the project to copy into.
6	In the Browser window, navigate to the location to import the Package into.
7	Click on the target Package and use your preferred Import access method to open the 'Import Package from Native/XMI File' dialog.
8	Select the appropriate options and filename.
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, the system creates a copy of the original Package

Model Exchange in XMI Format

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. Native file formats, on the other hand, are Enterprise Architect-specific formats for model exchange between various Enterprise Architect repositories.

XMI and Enterprise Architect

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. The system supports the XMI 1.1, 1.2, 2.1, 2.4, 2.4.2 and 2.5.1 specifications. XMI 1.1 provides support for UML 1.3, whereas XMI 2.x is used for UML 2.0 - 2.5.1.

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.

With XMI, model details can be exchanged between different UML tools and other tools that are capable of using XMI. Limited support for export to Rational Rose is provided using the Rose version of the XMI 1.1 specification, as implemented by Unisys for Rational products.

Packages can be exported from and imported into Enterprise Architect models, which greatly improves the flexibility and robustness of the models, helping Analysts and Modelers to externalize model elements in XMI for:

- Version Control
- Distributed development
- Post processing and
- Transferring Packages between models

When importing an XML file over an existing Package, by default all information in the current Package is deleted before the data is imported from the XML file. Please make sure you do not have important changes that you do not want to lose before importing from XML. Alternatively, use a Merge File to merge the information in the XMI file with the contents of the Package.

XML File Tasks

Tasks you might perform in importing and exporting Native/XMI file include:

- Setting XML Options - Native/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 Native/XMI file
- Importing from Native/XMI file
- 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)
- Importing a model from IBM Rational Rhapsody
- Importing a model from a MagicDraw MDZIP file

Notes

- When performing Enterprise Architect-to-Enterprise Architect transfers, ensure that XMI version 1.1, XMI version 2.1 or Enterprise Architect's Native XML/XEA is selected
- XMI 2.1 file exported by Enterprise Architect 7.0 or later might not be correctly imported by versions of Enterprise Architect earlier than 7.0
- Native XML file exported by Enterprise Architect version 15.0 or later cannot be imported by versions of Enterprise Architect earlier than 15.0
- Native XEA file exported by Enterprise Architect version 16.0 or later cannot be imported by versions of Enterprise Architect earlier than 16.0

Export to XMI

You use the 'Export Package to Native/XMI File' dialog to export a Package from your model to a file in one of several XMI (XML-based) or Native formats. The dialog, by default, allows exporting to XMI 1.1, XMI 2.1, Enterprise Architect's Native XML or Native XEA format. The exported 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

Select a Package in the Browser window, then:

Ribbon	Publish > Model Exchange > Export > Native Format Publish > Model Exchange > Export > XMI Format
Keyboard Shortcuts	Ctrl+Alt+E

Export Package to Native/XMI file

Option	Action
Package	Displays the name of the currently-selected Package. If this is not the required Package, click on the  button and select the correct Package.
Filename	Type in the file path of the XML/XEA file to be generated, or click on the  button and browse for the location.
Options	'Native XEA' - Select this option to export the selected Package to Enterprise Architect's Native XEA format 'Native XML' - Select this option to export the selected Package to Enterprise Architect's Native XML format (default if you have selected the 'Native Format' menu option) 'XMI 1.1' - Select this option to export the selected Package to XMI 1.1 (default if you have selected the 'XMI Format' menu option) 'XMI 2.1' - Select this option to export the selected Package to XMI 2.1
Generate Merge file using Baseline Version	This option is enabled when: 'XMI 1.1' is selected and - The selected Package has a baseline Click on the  button to open the 'Baselines' dialog and select the baseline. Once export is complete, the Merge File will be generated by comparing the selected Package with the selected baseline. The Merge File will be generated in the same location as the XMI file, with the same name as the XMI file but with the suffix _Merge.xml.

	This file is used to merge the contents of a Package with that of the XMI file, using the 'Merge Package with XMI' dialog.
Format XML Output	Select this check box to format the output. Note that Native XEA format doesn't support this option, so the option is grayed out for that format. You can set a default selection for 'Format XML Output' by selecting 'Start > Preferences > Preferences > XML Specifications > Format XMI Output'.
View XML	Click on this button to view the exported XML file. This button is disabled when 'Native > XEA' is selected.
Export	Click on this button to initiate the Package export.
Close	Click on this button to close this dialog.
Help	Click on this button to display this Help topic.
Progress	Monitor the progress of Package export.

Notes

- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Export XML' permission to export to XML
- Alternative Images used in the Package being exported are not included in the exported Native/XMI file, unless the 'Export Alternate Images' option is selected in the 'XML Specifications' section of the 'Preferences' 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 'Preferences' dialog
- XMI 2.1 files exported by Enterprise Architect Version 7.0 or later might not be correctly imported by versions of Enterprise Architect earlier than Version 7.0
- A Native XML file exported by Enterprise Architect Version 15.0 or later cannot be imported by versions of Enterprise Architect earlier than Version 15.0
- A Native XEA file exported by Enterprise Architect Version 16.0 or later cannot be imported by versions of Enterprise Architect earlier than Version 16.0

Import from XMI

You can copy Enterprise Architect model elements between models, for distributed development, manual Version Control and other benefits. You do so by importing the parent Package from an XMI (XML-based) or Native file in any of these formats:

- UML 1.3 (XMI 1.0)
- UML 1.3 (XMI 1.1)
- UML 1.4 (XMI 1.2)
- UML 2.x (XMI 2.1)
- MOF 1.3 (XMI 1.1)
- MOF 1.4 (XMI 1.2)
- Enterprise Architect's Native XEA/XML

Access

Select your target Package in the Browser window, then:

Ribbon	Publish > Model Exchange > Import > Native File or Publish > Model Exchange > Import > XMI File
--------	--

Import a Package From an XML File

Option	Action
Filename	Type the directory path and filename from which to import the XML or XEA file. Alternatively, click on the  button to display the file browser and search for the file. • If you selected the 'Import > Native File' option, the file-type field shows 'Native File (*.xea, *.xml)', filtering the listed files to those of that type • If you selected the 'Import > XMI File' option, the file-type field shows 'XMI File (*.xml, *.xmi)', filtering the listed files to those of that type • If you used the Ctrl+Alt+I shortcut, the file-type field shows 'Native/ XMI File (*.xea, *.xml, *.xmi)
Import Diagrams	Select the checkbox to import diagrams.
Strip GUIDs	Select the checkbox to remove Universal Identifier information from the file on import. By removing the GUID, you can import a Package twice into the same model; the second import requires the checkbox to be selected, to apply 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 _ import.log.
Import using Single Transaction	By default, this option will be disabled when importing into : • A Postgres repository or • Any repository over a Cloud connection Otherwise, this option defaults to the setting of the 'Import using single transaction' checkbox in the 'XML Specifications' page of the 'Preferences' dialog. If you prefer, you can change the option from the default setting. • If the checkbox is selected, the file is imported in one transaction.; this is not recommended for large imports. • If the checkbox is not selected the data items are imported separately; you can identify problem items without the whole import being blocked If the import is likely to encounter locking issues, or if you are importing a large file, deselect the checkbox.
Check if Package can be imported as Model Root	Select this checkbox if you need to enable the importer to check if the Package being imported from the Native/XMI file can be imported as a Model Root. If enabled, this check will be performed only when the Package being imported does not already exist in the model or when importing with the 'Strip GUIDs' option checked. If not enabled, then the Package from the Native/XMI file will be imported under the selected Package.
Import Discussions by	This option defaults to the setting of the 'XMI/VC Import Policy for Discussions' option on the 'General' page of the 'Manage Model Options' dialog. The option will be enabled if the option 'Allow user to override if prompted' is selected on the 'General' page of the 'Manage Model Options' dialog. If enabled, click on the drop-down arrow and select an option that determines how to import Discussions from XML : • Merging from XML - select this option to retain the existing Discussions in the model and merge those in Native/XMI file • Restoring from XML - select this option to delete the existing Discussions in the model and import those in Native/XMI file
Merge XMI	Click on this button to display two options for merging the content of the XMI file with the content of the selected Package. • 'Use Merge File' - Merge the contents of the XMI file with the contents of the Package using a Merge File • 'Direct Merge' - Merge the contents of the XMI file directly with the contents of the Package, without a Merge File
Treat Imported Datatypes as	If you are importing from Rose XMI 1.1, click on the drop-down arrow and select the datatypes to add to the model.
Baseline Package after Import	Select this checkbox to open the 'Create Baseline' dialog, through which to baseline the imported Package once the import is complete. Selecting this option also generates a Merge File when exporting a Package to XMI 1.1 using the 'Export Package to Native/XMI File' dialog.
Import	Click on this button to start the import.
Close	Click on this button to close the dialog.

Help	Click on this button to display this Help page.
Import Progress	Indicates the progress of the import.

Placeholders

When importing an Enterprise Architect model as XMI 1.1/2.1 or Native XML/XEA file with cross-Package references, you can use the Placeholder option to visually show on the imported diagram, any items that were referenced in the source model, but were not directly part of the export due to these Elements being located outside the exported Package-tree. For more details see the option *Create placeholders for missing External References during XMI 1.1/2.1 and Native File Import* in the *XML Specifications* Help topic.

Notes

- 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 Canonical XMI 2.1 files, BPMN 2.0 XML files, Rhapsody and MagicDraw models, and XML files exported by ArcGIS; select the appropriate 'Publish > Model Exchange > Import' ribbon menu option
- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Import XML' permission to import Packages from XML
- If you are working in a model on a Cloud connection, your ability to import from XMI might be blocked by a model-specific option set on the 'Cloud' page of the 'Manage Model Options' dialog; you can still import from XMI using a direct connection to the model
- If you are connected to a SQL Server repository, you require ALTER permission on the t_image table in Enterprise Architect, to be able to import model images
- When you import an XML file over an existing Package, ALL information in the current Package is deleted first, unless you use the Use Merge File button; before you import an XML file, please make sure you do not have important changes that you do not want to lose
- If you are importing an XMI 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 XMI file is being written; an error occurs if the UML_EA.DTD file is absent
- An XMI 2.1 file exported by Enterprise Architect Version 7.0 or later might not be correctly imported by versions of Enterprise Architect earlier than Version 7.0
- A Native XML file exported by Enterprise Architect Version 15.0 or later cannot be imported by versions of Enterprise Architect earlier than Version 15.0
- A Native XEA file exported by Enterprise Architect Version 16.0 or later cannot be imported by versions of Enterprise Architect earlier than Version 16.0

Merge with XMI Using Merge File

If you are working in a distributed environment, each modeler could be working on a separate file copied (as an XMI export/import) from the master model. At intervals you would want to bring each person's work back into the master model. One way of doing this is to firstly export each modeler's copy as an XMI file, and import and merge each file in turn with the corresponding Package in the master model.

The logical next step would be to again export the master model as an XMI file so that each modeler could then import that into their copy to synchronize their model with the latest version of the whole team's work. However, if any of the team have continued to work on their copy, they can also merge the 'master' XMI file with their copy so that they do not overwrite their latest work.

Access

Select your target Package in the Browser window, then use one of the methods outlined here to open the 'Merge Package with XMI' dialog.

Ribbon	Publish > Model Exchange > Import > Merge Model XMI into Current Package
Keyboard Shortcuts	Ctrl+Alt+I : Merge XMI > Use Merge File

Merge Package options

Option	Action
Package	Displays the name of the currently-selected Package, the contents of which will be merged with the imported XMI file. If this is not the required Package, click on the  button and select the correct Package.
XMI File	Type in the directory path and filename of the XMI file, or click on the Browse button and locate the file.
Merge File	Type in or browse to the directory path and filename of the Merge File, generated using the 'Export Package to Native/XMI File' dialog. The field defaults to show the Merge File in the same location as the XMI file.
Baseline Package before Merge	Select the checkbox to show the 'Create Baseline' dialog, to baseline the selected Package before merging the XMI file. You can use this baseline to roll back to the content of the pre-merged Package, if required.
Compare Package with Baseline after Merge	Select the checkbox to automatically compare the Package with the Baseline once the merge is complete. This checkbox is enabled when the 'Baseline Package before Merge' option has been selected.
Merge	Click on this button to merge the Package and its contents with the contents of the

	XMI file.
Close	Click on this button to close the 'Merge Package with XMI' dialog.
Help	Click on this button to display this Help page.

Merge with XMI

You have two options for importing a Package into a model from XMI, where that Package already exists in the model:

- Merging the Package directly, as documented here, where Enterprise Architect simply merges the differences between the XMI file and the existing Package in the model
- Using a Merge File, as documented in the *Merge Package with XMI Using Merge File* Help topic, where Enterprise Architect deletes the existing Package from the model and imports the new Package information from XMI

Access

Select your target Package in the Browser window, then use one of the methods outlined here to open the 'Import Package from Native/XMI File' dialog.

On the 'Import Package from Native/XMI File' dialog, click on the Merge XMI button, then choose the option 'Direct Merge'.

Ribbon	Publish > Model Exchange > Import > Native File (or Import > XMI File) : Merge XMI > Direct Merge
Keyboard Shortcuts	Ctrl+Alt+I : Merge XMI > Direct Merge

Merge Package Options

Option	Action
Package	Displays the name of the currently-selected Package, the contents of which will be merged with the imported XMI file. If this is not the required Package, click on the  button and select the correct Package.
XMI File	Type in the directory path and filename of the XMI file, or click on the  button and locate the file.
Baseline Package before Merge	Select this checkbox to open the 'Create Baseline' dialog, to create a baseline of the selected Package before merging the XMI file. You can use this baseline to roll back to the content of the pre-merged Package, if necessary.
Compare Baseline with Package after Merge	Select this checkbox to automatically compare the Package with the Baseline once the merge is complete. The checkbox is enabled when the 'Baseline Package before Merge' checkbox is selected.
Merge	Click on this button to begin the merge of the Package content with the XMI file content.
Close	Click on this button to close the 'Merge Package with XMI' dialog.

Help	Click on this button to display this Help page.
------	---

Import EMX/UML2 Files

In Rational Software Architect (RSA) you can 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 Enterprise Architect releases earlier than Release 7.0, each EMX file is treated as a separate model and hence does not allow for cross-references between them. Since 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

Select your target Package in the Browser window, then use the method outlined here to open the 'Import Package from EMX/UML2' dialog.

Ribbon	Publish > Model Exchange > Import > EMX/UML2
--------	--

Import Files

Option	Action
Package	Displays the name of the currently-selected Package. If this is not the required Package, click on the  button and select the correct Package.
Directory	Click on the  button. The 'Select Import EMX/UML2 File(s)' dialog displays, from which you can select multiple files. 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 Package from EMX/UML2' dialog.
Selected Files	Lists the selected *.emx/*.uml2 files.
Write Log File	Select this checkbox to write a log of the import activity (recommended). The log file is saved in the directory from which the files are being imported, with the name import.log.
Import	Click on this button to import the *.emx/*.uml2 files.
Close	Click on this button to abort the import and close the dialog.
Help	Click on this button to display this Help topic.

Import Progress	Monitor the progress of the import.
-----------------	-------------------------------------

Import a Rhapsody Model

If you have developed a UML model in IBM Rational Rhapsody, you can import it into an Enterprise Architect project.

Each Rhapsody project contains one Project file (with the file extension .rpy), which is associated with a folder with the same name, 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 could also contain a file named *filesTable.dat*, which lists all the files related to the Rhapsody project.

Access

Select your target Package in the Browser window, then use the method outlined here to open the 'Import Model from Rhapsody' dialog.

Ribbon	Publish > Model Exchange > Import > Rhapsody
--------	--

Import UML Model from Rhapsody Project Files

Option	Action
Package	Displays the name of the currently-selected Package. If this is not the required Package, click on the  button and select the correct Package.
Rhapsody Project	Click on the  button. The 'Select Import Rhapsody Project' dialog displays. Select a Rhapsody Project file, then click on the Open button to return to the 'Import Model From Rhapsody' dialog.
Rhapsody Files	Shows the name of the file located in the associated project folder.
Import	Click on this button to import the Rhapsody model into Enterprise Architect.
Close	Click on this button to abort the import and close the dialog.
Help	Click on this button to display this Help topic.
Import Progress	Indicates the progress of the import.

Import a MagicDraw MDZIP File

Enterprise Architect supports importing UML models from MDZIP files generated by MagicDraw 17.0.1 and above.

Access

Select your target Package in the Browser window, then use the method outlined here to open the 'Import from MDZIP File(s)' dialog.

Ribbon	Publish > Model Exchange > Import > MDZIP File
--------	--

Import UML Model from MDZIP File(s)

Option	Action
Package	Displays the name of the currently-selected Package. If this is not the required Package, click on the  button and select the correct Package.
Filename	Click on the  button. The 'Select MDZIP File(s)' dialog displays, from which you can select multiple files. Select the required MagicDraw MDZIP file or files (use Shift+click or Ctrl+click to select several files), then click on the Open button to return to the 'Import From MDZIP File(s)' dialog.
Selected File(s)	Lists the *.MDZIP files you have selected for import.
Write Log File	Select this checkbox to write a log of import activity. The log file is saved in the directory from which the files are being imported, with the name import.log.
Import	Click on this button to import the MDZIP file(s) into Enterprise Architect.
Close	Click on this button to close this dialog.
Help	Click on this button to display this Help page.
Import Progress	Indicates the progress of the import.

Notes

- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Import

XMI' permission to import Packages from XML

Limitations of XMI

Whilst XMI is a valuable means of defining a UML model in a common format, it is relatively limited in the amount of additional information it can tolerate using the standard syntax.

Notes on Enterprise Architect

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. Enterprise Architect can both generate and read:

- 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.x format

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.

Enterprise Architect exclusively uses XMI 1.1 for round-tripping models using Version Control.

Notes for Exporting to Rose and Other Tools

There are discrepancies in the Unisys/Rose implementation with regard to typographic errors and slightly different syntax to the official XMI 1.1 specification, so problems might occur. The way Packages are arranged in different models can impact successful import into other systems; experimentation is the only work around for this problem.

Some parts of the XMI import/export process do not work as expected in products such as Rational Rose; for example, Note Links are not supported, and State Operations import but do not appear in diagrams. Rational Rose only supports import of a full project, not a single Package.

For best results, it is recommended that you keep the model elements to export to Rational Rose simple and that they conform as closely as possible to the UML 1.3 specification.

The UML DTD

When you import or export Packages from an Enterprise Architect model to XMI, the import or export process can be validated using a Data Type Definition (DTD). Several DTDs for XMI/UML exist; the OMG defines a standard UML1.3 DTD for use in XMI 1.1, and Enterprise Architect uses an extension of this with some additional element extensions for non-standard UML types, such as testing details.

The XML parser uses the DTD to validate the correctness of the model and to check that no syntactical errors have occurred. It is always best to use a DTD when moving Packages between Enterprise Architect models, as it ensures correctness of the XMI output and prevents attempted imports of incorrect XML.

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. If the parser cannot find the DTD, it records an error and aborts processing; the UML_EA.DTD file must be in the current XML output path (generated by default).

Model Exchange with Custom Package Sets

You generally use a Model Document element to create a document report containing information from separate Packages in your model. However, you can use the same concept, structure and processes to generate an Enterprise Architect Native XML file for export to another model. This helps you to select Packages from any part of the model and transport them to another model without having to process them individually or create special structures to combine them. You could use this to, for example, define a model subset that can be included in a restricted-user WebEA view.

You create a Model Document element, add the individual Packages to it, and generate the XML from that element to an external file location. The XML will contain all the Packages (and their child structures) specified in the Model Document element. You can then use the normal Import XML facilities to import the file into another model.

You can generate an XML file from a Model Document element set up to generate a document report from a Package list; the XML generator ignores the Tagged Values of the element. You could generate a report to accompany the file, to show its content.

Set up the Model Document

Work through these procedures:

- Create a Model Document element on a diagram (as described in the *Create Model Document* Help topic); you don't need to set any Tagged Values
- Drag the required Packages into it (as described in the *Add Packages to Model Document* Help topic); you don't use a model search to populate the element
- If necessary, organize the Packages into an appropriate sequence (as described in the *Document Order* Help topic, *Change Sequence of Packages...* section)

Generate XML

Step	Action
1	When you have set up the Model Document element, right-click on it in its parent diagram and click on the 'Documentation Generate XML' option.
2	On the 'Generate Model Document XML' dialog, type in the file path of the XML file to be generated, or click on the  button and browse for the location. Click on the Generate button to generate the XML into the file.
3	The XML Generator works through the defined content of the Model Document element, pulls in the information from the listed Packages and generates a Native XML file, showing the progress of the operation in the Progress field. Click on the Close button. You - or another user - can now import the XML file into a different model using the 'Import Package from Native/XMI File' dialog.

Notes

- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Export

XML' permission to export to XML

- The generated XML file cannot be imported into a version of Enterprise Architect earlier than version 15.0
- A Log file for the Model Document being exported is not generated unless the 'Write Log' option is selected in the 'XML Specifications' section of the 'Preferences' dialog
- When you import the XML file into a model, Enterprise Architect will create a Package (with the same name as the Model Document element) that will act as a container for the Packages specified in the Model Document element

Transfer a Complete Repository

In normal system maintenance, it is likely that at some point you need to relocate data between storage systems. The purpose of the Complete Project Transfer feature is to completely overwrite everything in the target repository with the contents of the source repository. The main advantage to this feature is that any supported repository type can be transferred to any other repository type.

In any edition of Enterprise Architect, you might need to transfer all records from one project file to another (recommended after serious network crash or repeated database corruption). Project files can be either .qea (SQLite), .feap (Firebird) or .eap (Access) files, and you can transfer from one file type to the other.

In the Corporate, Unified and Ultimate Editions of Enterprise Architect, you might need to:

- Populate a new project file from an EABase template project, or another project you want to duplicate (using one of the 'File to...' options)
- Transfer an existing project file to a DBMS repository such as SQL Server or MySQL
- Transfer a DBMS repository to an existing project file
- Move a repository from one DBMS to another (including moving between servers using the same DBMS)
- Copy all records from a JET 3.5 to JET 4 (Access 2000 or XP) repository - or back the other way

You can also transfer a .QEA file (Release 16.0 and higher, 64 bit) to a .FEAP file (Release 15.2 and earlier) for backwards compatibility. See the *Transfer a QEA/QEAX Repository to FEAP Format* Help topic.

Access

Ribbon	Settings > Model > Transfer > Full Project Transfer via Connection
--------	--

Perform a Project Data Transfer

Step	Action
1	Run a Check Project Data Integrity against the source repository and correct any problems.
2	If the target Project contains data then generally it is suggested to take a backup of that project to ensure that you can recover any important information it contains. For a File based repository this can be achieved by making a copy of the Project file, but if the target is DBMS based then the official backup utility of the DBMS should be used.
3	Click on the radio button for the required transfer type: • File to File • DBMS to File • File to DBMS • DBMS to DBMS
4	In the 'Source Project' and 'Target Project' fields, browse for and select the connection strings for the Source and Target projects. You might be prompted to select the database connection to use, either the 'Open Server Connection' dialog (Native Connection) or 'Data Link Properties' dialog (ODBC Connection).

5	Check the 'Reset GUIDs' option if all Packages/diagrams/elements should be given a new value in the target repository. If the target repository is going to be used as a new repository and is completely independently from the source then we would suggest that the 'Reset GUIDs' option should be checked, but if the target repository is a backup or you want a clone of your source repository then 'Reset GUIDs' should be unchecked.
6	If you want to capture the steps that occur during the transfer (this is generally recommended), select the 'Logfile' checkbox and browse for the appropriate log file location.
7	Click on the Transfer button. Note: if the target repository has user security enable then you will be prompted for the credentials of a user with the <i>Transfer Data</i> permission, this is a security measure to ensure that unauthorized users don't over write a repository.
8	It is good practice to do a Project Compare after this process to verify that all records are written.

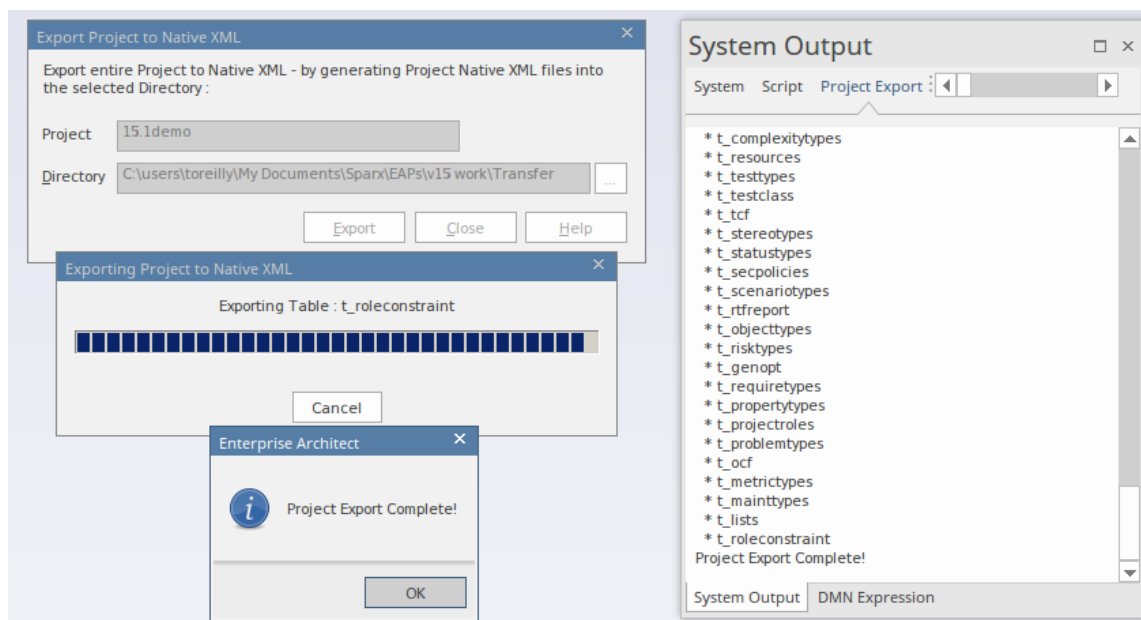
Notes

- When the target repository is a MySQL or SQLServer based repository, the database user requires more than just read and write access to the database. MySQL users require ALTER and DROP permissions while SQL Server users require the role DB_DLLADMIN. If the database user does not have the necessary permissions the process will display an error message and be unable to continue
- When performing a .eap file to DBMS transfer, if the .eap file has replication enabled, the replication must be removed before performing the transfer
- When transferring a model to a DBMS, you cannot transfer the data into an empty database - the tables must already exist; if they do not, run the corresponding script supplied by Sparx Systems to create the required tables
- When transferring a model into a project file, similarly, the target project file must already exist, to provide the data structures to transfer data into
- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Transfer Data' permission to transfer project data between repositories
- You cannot move a model from a source .eap file from an Enterprise Architect version earlier than 3.5.0

Transfer a Repository in Native Format

We live in an age of information and digital disruption, where individuals and companies have an almost insatiable thirst for information and knowledge. While it is possible to access the valuable corporate modeling assets stored in an Enterprise Architect repository from any device in any location, there still remain scenarios where a modeler will want to extract or exchange the information stored in the repository. Enterprise Architect is compliant with standards and has always provided a facility to export the models to the standards-based XMI format, but it now also provides a Native XML format export and import facility for the transfer of an entire project's data. This is a two step process that involves:

1. Opening the source project in Enterprise Architect and exporting it to Native XML using the 'Export Project to Native XML' dialog.
2. Opening the target project in Enterprise Architect and importing the Native XML files exported in the first step, using the 'Import Project from Native XML' dialog.



Regardless of the size of the repository, the model content - including all elements, connectors, documents and reference data - can be conveniently exported and exchanged between repositories at Internet speed. Exporting the data from one repository and importing it into another is fast and can conveniently be used with any type of repository connection (Cloud, Server or file system) and any type of repository (RDBMS or file). There is a range of other uses for the exchange of repository content, including such things as backing-up repositories, record keeping or exchange with other tools.

The feature is particularly useful for Cloud-based models where a modeler does not have direct access to the Cloud hosting environment and the data sources are protected by private subnet security. The Native XML format export and import can be used effectively as a project transfer option for Cloud repositories. The data transfers would typically be secured with secure socket layer connections, preventing the data from being compromised in either the exporting or importing phases of a transfer.

Access

Ribbon	Settings > Model > Transfer > Full Project Transfer via Native XML > Export Complete Project Settings > Model > Transfer > Full Project Transfer via Native XML > Import Complete Project
--------	--

Export Project Data to Native XML files

Use the 'Export Project to Native XML' dialog to export the current project to Native XML. The export process will generate multiple Native XML files, representing data of the project.

Option	Action
Project	Displays the name of the current project (selected in the Browser window).
Directory	Type in the path of the directory in the file system into which the Native XML files, representing the project data, are to be generated; alternatively, click on the  button and browse for the directory location. Note: • If the directory already contains Project Transfer specific XML files (from a previous Project Transfer export), all these files will be deleted prior to exporting project data • If the directory contains non-Project Transfer specific files, you will be prompted to create a sub-directory to export project data
Export	Click on this button to initiate the Native XML export. When the export is successfully completed, the specified directory will contain multiple Native XML files representing the project data. Note: This data will not be exported as a part of the Project Transfer: • Audit Logs • Image and Image-maps of diagrams • HTML Pages of Linked Documents
Close	Click on this button to close the 'Export Project to Native XML' dialog.

Import Project Data from Native XML files

Use the 'Import Project from Native XML' dialog to import Native XML files generated by the export process into the current (target) project. The contents of the current project will be deleted prior to importing the Native XML files; if necessary, take a backup of the current project prior to importing the files.

Option	Action
Project	Displays the name of the current project.
Directory	Type in the path of the directory in the file system into which the Native XML files, representing the project data, were generated by the <i>Export Project Data to Native XML files</i> process; alternatively, click on the  button and browse for the directory location.
Import	Click on this button to initiate the Native XML import. The import progress will be displayed in the 'Project Import' tab in the System Output window.

Close	Click on this button to close the 'Import Project from Native XML' dialog.
-------	--

Notes

- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Transfer Data' permission and:
 - 'Export XMI' permission to export project data to Native XML
 - 'Import XMI' permission to import project data from Native XML
- Native XML files, representing project data, cannot be imported using the 'Import Package from Native/XMI File' dialog
- If you are connected to a SQL Server repository, you require ALTER permission on the t_image table in Enterprise Architect to be able to import model images
- Project can also be exported to Native XML using the Automation Interface function ExportProjectXML()

Native File Formats

Native XEA and Native XML are Enterprise Architect-specific file formats for model exchange between various Enterprise Architect repositories. Native file formats are, therefore, a mechanism for round-tripping models (that is, performing Enterprise Architect-to-Enterprise Architect transfers). Compared to XMI 1.1/2.1, these file formats have a structure that closely mimics the Enterprise Architect repository, so exporting to Native file formats are considerably faster compared to exporting to XMI 1.1/2.1.

By default, when importing an XMI 1.1/2.1 file over an existing Package, all information in the current Package is deleted from the model before the data is imported from the XMI file. With a Native file import, the existing Package will not be deleted from the model, but will instead be updated with data from the Native file. Therefore, importing a Native file will also be considerably faster than importing from XMI 1.1/2.1.

When exporting a Package to Native file format, the Package being exported will become the root Package in the file. The children of the Package being exported (that is, its elements, diagrams, connectors and so on) will be exported under the root Package in the file. When importing this file into a model, the Native File Importer will check if:

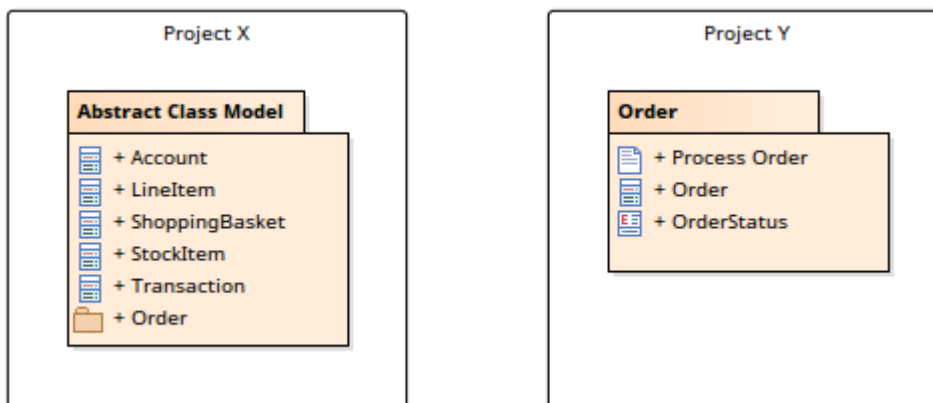
- The root Package in file exists in the model
- Any of the children of the root Package in file exist in the model

If the target model does not contain the root Package in file or its children, then the contents of the file will be created as new items in the model.

If the target model contains the root Package in Native file and its children, then the existing items in the model will be updated with the contents of the Native file.

Importing a Native file to a model that contains some of the imported items

Your model might contain at least one Package that is also used in other models, but not necessarily in the same context. For example, consider the two projects shown here:



Package 'Order' exists in two separate projects, 'Project X' and 'Project Y'. In 'Project X' the 'Order' Package is available as a child of Package 'Abstract Class Model', whereas in 'Project Y' it is a View Package.

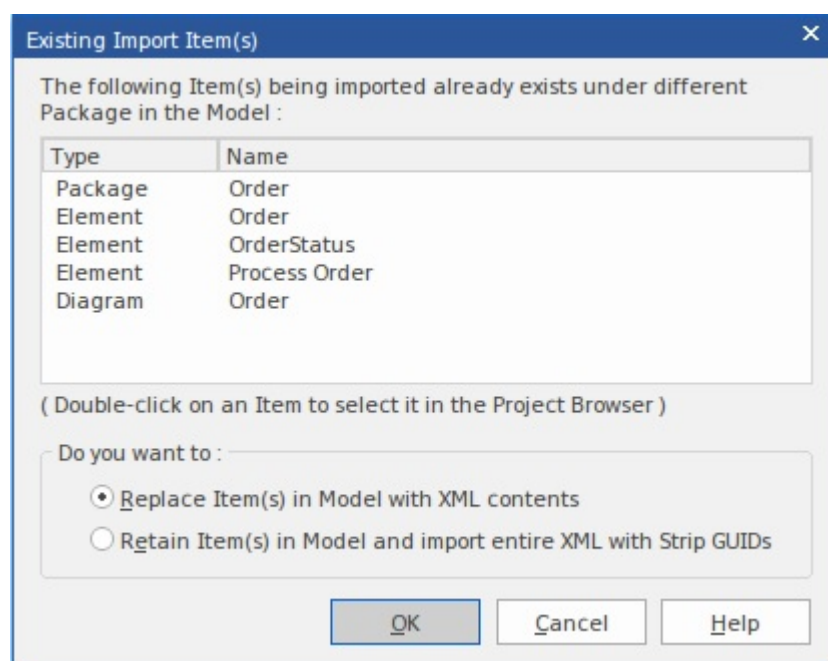
When you export the Package 'Abstract Class Model' from 'Project X' to Native file, the file will contain:

- 'Abstract Class Model' as its root Package and
- All information on 'Abstract Class Model' and its children (including Package 'Order')

When this file is imported into 'Project Y', the Native File Importer will determine that:

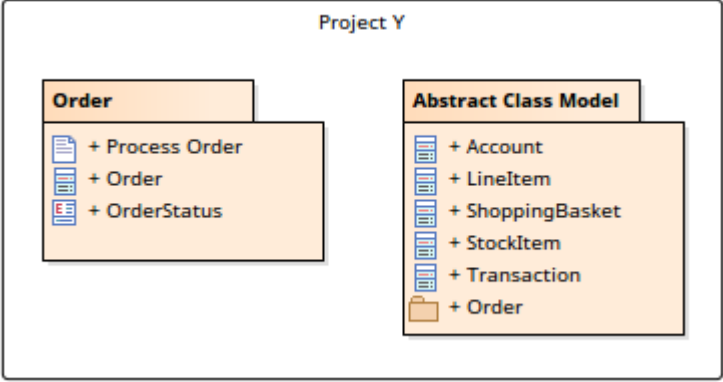
- The root Package in Native file, 'Abstract Class Model', does not exist in the model
- The child Package 'Order' does exist in the model

In such a scenario, where the root Package in Native file does not exist in the model but its children do, the 'Existing Import Item(s)' dialog will be displayed, which will list the items in Native file that already exist in the model:



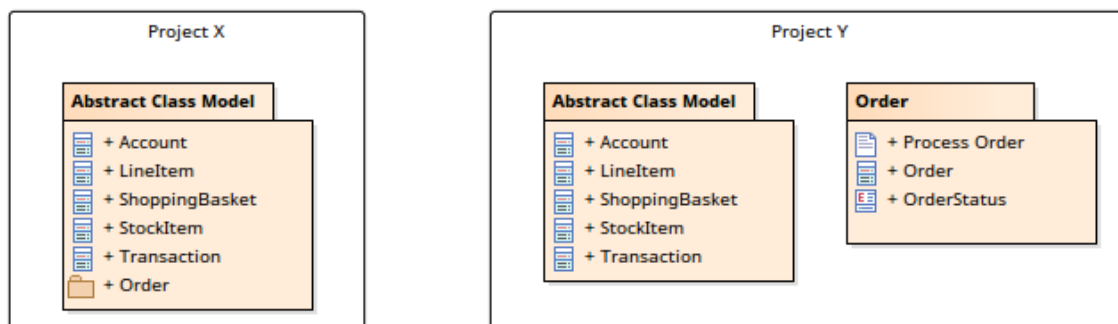
Actions

Action	Detail
Double-click on an item in the list	Selects the item in the Browser window.
Replace Item(s) in Model with Native file contents	Select this radio button to: - Delete these items from the model and - Import them from the Native file In our example, after import 'Project Y' will have this content:
Retain Item(s) in Model and import entire Native file with Strip GUIDs	Select this option to: - Retain (that is, not modify) these existing Item(s) in the model - Import the entire Native file with GUIDs stripped from them In our example, after import, 'Project Y' will have this content:

	 As you can see, two copies of Package 'Order' will exist in 'Project Y' after import. The View Package 'Order' that existed in the model before import will remain unchanged. The Package 'Order' from the Native file will be imported as a new Package under the imported 'Abstract Class Model' Package.
Cancel	Cancel the Native file import.

Importing Native files into Models that contain the root Package and children, in a different location

Consider the combination of Projects and Packages shown here:



Packages 'Abstract Class Model' and 'Order' exist in two separate projects, 'Project X' and 'Project Y'. In 'Project X', 'Order' exists as a child of Package 'Abstract Class Model', whereas in 'Project Y' they are both peer Packages.

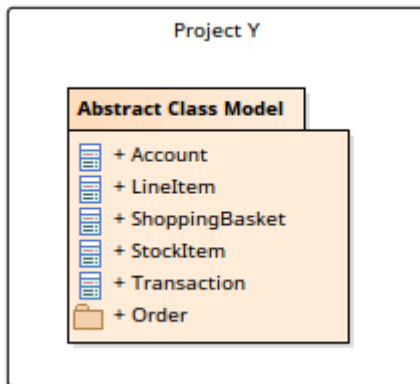
When Package 'Abstract Class Model' is exported from 'Project X' to Native file, the file will contain:

- 'Abstract Class Model' as its root Package
- All information about 'Abstract Class Model' and its children (including Package 'Order')

When this file is imported into 'Project Y', the Native File Importer determines that:

- The root Package in file, 'Abstract Class Model', exists in the model
- The child Package 'Order' exists in the model
- The root Package and its children are in different locations in the model

In such a scenario, where the root Package in file and its children exist in the model (albeit in different locations), the Importer will replicate the Native file Package structure; that is, it will automatically move Package 'Order' under Package 'Abstract Class Model' in the model. The 'Existing Import Item(s)' dialog will not be displayed. After import, 'Project Y' will have the contents shown:



Notes

- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you need to have:
 - 'Export XMI' permission to export Packages to Native file format
 - 'Import XMI' permission to import Packages from Native file format
- Native XML files exported by Enterprise Architect version 15.0 or later cannot be imported by versions of Enterprise Architect earlier than 15.0
- Native XEA files exported by Enterprise Architect version 16.0 or later cannot be imported by versions of Enterprise Architect earlier than 16.0
- Native XML files exported using the 'Export to Native/XMI File' dialog will not be formatted (that is, 'pretty-printed'), even if the 'Format XMI Output' option is enabled in the 'XML Specifications' option-group of the 'Preferences' dialog
- If you are connected to a SQL Server repository, you require ALTER permission on the t_image table in Enterprise Architect, to be able to import model images

Migrate an EAP/EAPX File to QEA File Format

An EAP file is a Microsoft Jet 3.5 database, and an EAPX file is a Microsoft Jet 4.0 database. The Jet database engine is available only in 32 bit configuration. Therefore, the 64 bit version of Enterprise Architect Release 16.0 and above will not be able to open an EAP or EAPX file. From Enterprise Architect 16.0, to keep using the data in an EAP or EAPX file you must transfer the contents of the file to another project that the 64 bit version can access. This project can be a file (such as a .QEA or Firebird file) or a database repository (such as a SQL Server or MySQL database).

Enterprise Architect also has a quick-transfer option to specifically convert a .eap file into a .qea file, faster than the more general option here. This option is described in the third table in this topic.

In a transfer all records in the target repository are overwritten, so if you are transferring to an existing target project take a backup of that project to ensure that you can recover any important information it already contains.

Access

Ribbon	Settings > Model > Transfer > Migrate EAP/EAPX Project to QEA
--------	---

Perform an EAP/EAPX Project Transfer

Step	Action
1	In the 'Source Project' field, click on the  button and browse for and select the name of the source EAP or EAPX file to transfer.
2	In the 'Type' field, click on the drop-down arrow and select the target project type. This can be either a file (.QEA - the default - or .FEAP) or a database repository (such as SQL Server or MySQL) through the Cloud, native connection or ODBC Driver connection.

	Depending on the type, the 'Project' field will enable you browse for and select the target project: • File - select a QEA, QEAX or FEAP file • Database - Cloud Connection - connect to a database repository using the 'Cloud Connection' dialog • Database - Native Connection - connect to a database repository using the 'Open Server Connection' dialog • Database - ODBC Connection - connect to a database repository using the 'Data Link Properties' dialog (not available in the 64 bit version on Linux)
3	Click on the Transfer button. The progress and completion of the transfer are displayed in the 'Progress' panel.

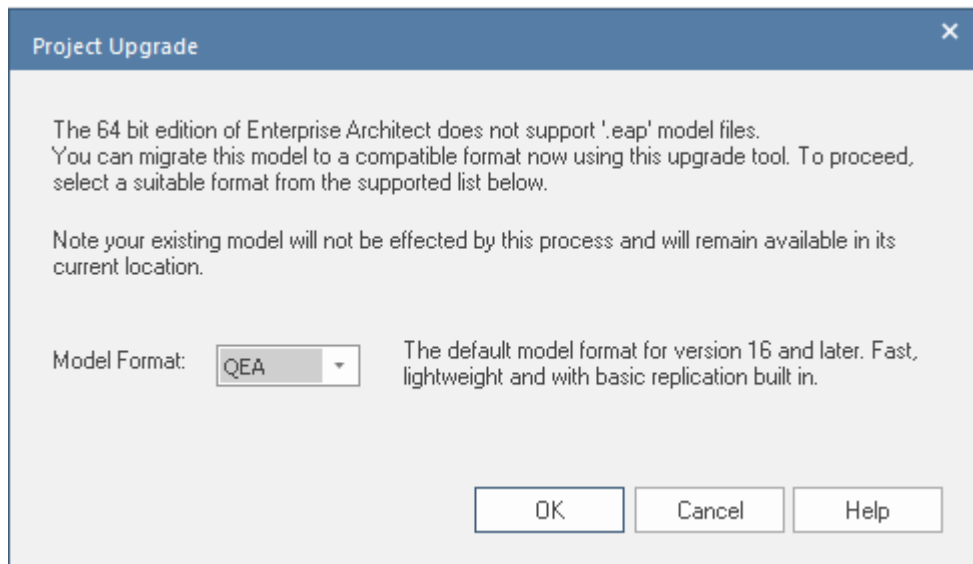
Notes

- When transferring a model to a DBMS, you cannot transfer the data into an empty database - the tables must already exist; if they do not, run the corresponding script supplied by Sparx Systems to create the required tables
- If the target project file exists, its contents will be deleted and Enterprise Architect will populate the file with the contents of the source project
- if the target project file does not exist, Enterprise Architect will create it and then populate it with the contents of the source project
- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Transfer Data' permission to transfer project data

Project Upgrade to QEA

With the introduction of 64 bit Enterprise Architect, we are migrating away from legacy EAP file repositories to the new default format of QEA or QEAX file repositories (as well as an option to use FEAP). This is primarily due to the lack of 64 bit support in the EAP architecture.

We have created an easy to use migration tool that will assist you in this process, the 'Project Upgrade' dialog, which is automatically displayed when it is needed.



Specify the Format

Simply click on the 'File Format' drop-down arrow and select the file format in which to create the new version of the repository. Then click on the OK button.

When you select a format, a brief description displays to the right of the field, similar to these:

- QEA - The default model format for version 16 and later; fast, lightweight and with basic replication built in
- QEAX - The same as QEA but with support for small work groups using a shared file
- FEAP - A legacy format compatible with version 15 and version 16

If you have Microsoft Access 2007 (or greater) runtime installed under Windows, you can also update the legacy EAP format to EAPX, and the EAPX format is also listed in the drop-down.

- EAPX - An MS Access format compatible with version 15 and 16/32; 16/64 requires special runtimes to be installed

Results

In a few moments, Enterprise Architect will finish the migration and open the file, and you can start browsing your upgraded model.

The new version of the model is created with the same name in the same folder as the original EAP file, which remains unchanged. If a model matching the name and new model type is found in the same location, a prompt displays to either proceed or exit the current upgrade process.

Proceeding with the upgrade when the model already exists will delete the current contents of the model and replace it with the content of the EAP file.

Transfer a QEA/QEAX Repository to FEAP Format

It is possible that at some point you might need to share your SQLite QEA file with a team member who is using an older version of Enterprise Architect (version 15.2 or earlier). However, older versions of Enterprise Architect cannot open a QEA file. In such situations, you can use the 'Copy QEA File to Firebird (FEAP)' option to transfer all the records from the QEA file to a Firebird file that can be opened in the older versions. This option is available in the Corporate, Unified and Ultimate Editions of Enterprise Architect.

All records in the target repository are overwritten in a transfer. So if you are transferring to an existing target Firebird project, take a backup of that project to ensure that you can recover any important information it already contains.

Access

Ribbon	Settings > Model > Transfer > Copy QEA Project to Firebird (FEAP)
--------	---

Perform a Project Data Transfer

Step	Action
1	Note that in the 'Transfer Type' panel the 'QEA to FEAP' radio button defaults to selected. In the 'Source Project' field, browse for and select the name of the source QEA project.
2	In the 'Target Project' field, browse for and select the name of the target Firebird project.

3	If you want to capture the transfer in a log file, select the 'Logfile' checkbox and browse for the appropriate log file location.
4	Click on the Transfer button, and monitor the progress of the transfer in the 'Progress' bar.
5	When the process is complete, it is good practice to do a Project Compare to verify that all records are written.

Notes

- This option is available in both 32 bit and 64 bit versions of Enterprise Architect Release 16.0
- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Transfer Data' permission to transfer project data

Model Publishing to Various Formats

The Publish Model Package facility is used to export a Package to a file in one of several XML formats, including XMI 1.1, XMI 2.1, Enterprise Architect's Native formats, Ecore, ArcGIS and BPMN 2.0 XML; for example, you might:

- Export BPMN 2.0 models to BPMN 2.0 XML/XPDL2.2, or
- Export Enterprise Architect model elements to Rational Rose and other tools that implement the:
 - UML 1.3 XMI 1.1 / XMI 1.0 standard
 - UML1.4 XMI 1.2 standard, or
 - UML 2.x XMI 2.x standard

Access

Select a Package in the Browser window, then:

Ribbon	Publish > Model Exchange > Publish As...
Keyboard Shortcuts	Ctrl+Alt+E : Publish

Publish Model Package options

Option	Action
Package	Displays the name of the currently-selected Package. If this is not the required Package, click on the  button and select the correct Package.
Filename	Type in or browse for () the file path into which the XML file is to be generated.
XML Type	Click on the XML/XMI version to which to export the Package. • Select 'UML 2.1' to use XMI 2.1 in round-tripping a model • Select 'UML 2.0, 2.1.1, 2.1.2, 2.2, 2.3, 2.4.1', '2.5' or '2.5.1' if exporting to a tool that requires that specific version of the XMI • Select 'Native XML' or 'Native XEA' to export to Enterprise Architect's Native format • Select 'Ecore' to export into a .ecore xmi file • Select 'BPMN2.0 XML' to export into a BPMN2.0 xmi file • Select 'XPDL 2.2' to export into an XPDL 2.2 xmi file • Select 'ArcGIS' to export into an ArcGIS xmi file The selected version might render some of the next eight options unavailable.
Export Diagrams	Select this checkbox to export all the diagrams in the selected Package.
Export Alternate Images	Select this checkbox to export the alternative images used in the diagrams in the

	selected Package.
Format XML Output	Select this checkbox to format the output into readable XML (this takes a few more seconds at the end of the run).
Write Log File	Select this checkbox to write a log of the export activity (recommended). The log file is saved to the directory into which the XML file is exported.
Use DTD	Select this checkbox to 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	Select this checkbox to exclude tool-specific information from the export.
Unisys/Rose Format	Select this checkbox to export in Unisys/Rose compatible XMI format.
Generate Diagram Images	Select this checkbox to generate the exported diagrams in the format you select from the 'Format' drop-down list (Metafile, Bitmap, GIF PNG, JPG, SVG). The generated diagrams are saved in a Package called Images in the directory into which the XML file is exported.
Stylesheet	(Optional) Click on the drop-down arrow and select an XSL Stylesheet to post-process the XML content before saving to file. 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' tab of the Browser window.
View XML	Click on this button to view the exported XML file.
Export	Click on this button to initiate the XML export.
Close	Click on this button to abort the instructions and close this dialog.
Help	Click on this button to display this Help topic.
Progress	Monitor the progress of the XML export.

Notes

- When exporting and importing with XMI 1.0 with Enterprise Architect, some loss of data occurs due to the limitations of XMI 1.0
- Exporting to BPMN 2.0 XML and XPD 2.2 is supported in Professional, Corporate, Unified and Ultimate Editions of Enterprise Architect
- In the Corporate, Unified 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 XMI 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

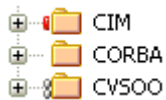
XMI export

Using Controlled Packages

A Controlled Package is a Package configured to save and load in Native/XMI format to a named file. Native formats are Enterprise Architect-specific file formats while XMI is UML compliant, with Enterprise Architect extensions to support diagrams and additional model elements. Controlled Packages are a useful means of 'externalizing' parts of an Enterprise Architect model.

When configured to use XML, the output 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.

Controlled Packages appear in the Browser window with a small colored rectangle to the left of the Package icon, as shown for the CIM Package:



Using Controlled Packages you can support:

- Widely distributed development by having team members submit Packages in Native/XMI format 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
Using XMI this way, you can 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
- Import and export of model elements between different models
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

Notes

- If you are importing an XMI 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 XMI file is being written; an error occurs if the UML_EA.DTD file is absent
- When you select to apply a DTD during an XMI 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 XMI export

Controlled Package Menu

If you are working on a Package that is not under Version Control, you can select certain options on the 'Controlled Package' context menu.

(Further options are available from the 'Model Exchange' panel of the Publish ribbon, and for a Package that is directly under Version Control you use the 'Version Control' panel of the Settings ribbon.)

Access

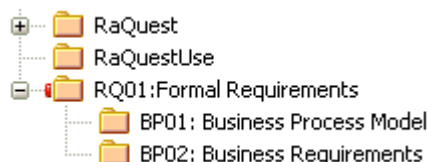
Context Menu	Browser window > Right-click on Package > Package Control
--------------	---

Package Control Options

Option	Action
Lock Package	Select this option to apply a lock on the selected Package. (This option is only displayed when user security is enabled in User/Group locking mode.)
Apply/Release User Lock	Select this option to apply or release a user lock on the selected Package. (This option is only displayed when user security is enabled in 'Require User Lock to Edit' locking mode)
Set Visibility Level	This menu item is only available in a model that has been set up for Visibility Levels, and when it is being accessed via a Pro Cloud server connection. It is primarily used by System Administrators to manage user access to Packages within the model. You use this option to set a Visibility Level on a Package, which applies to the Package and its child structures. The Visibility Level has a matching user access level set in User Security. A user with a given access level can only see, access and update Packages with the equivalent Visibility Level or lower. Select from the list of values, which start at 0 (no restriction on access) and can go up to 5 (for Packages requiring high security of access). You will only be able to set Visibility Levels at or below the access level to which you have been assigned. To remove a Visibility Level from a Package (and all its children), simply set the Visibility Level back to zero. Any user can then view and update the Package (within the limitations of any user security controls on their access to <i>functions</i>).
Import a Model Branch	Select this option to retrieve a model branch and import it into either the source model or another model.
Version Control Settings	Select this option to display the 'Version Control Settings' dialog.

Configure Packages

In order to configure a Package to save and load in Native/XMI format to a named file, you use the 'Package Control Options' dialog to flag the Package as being controlled and to set a number of options. These options include setting the filename to save to/load from, the type of Native/XMI to import and export, and whether to export and/or import the Native/XMI in a batch process. When the Package is configured, this is indicated by a small colored rectangle on the left of the 'Package' icon in the Browser window, as shown for the RQ01: Formal Requirements Package.



Access

Select a Package in the Browser window, then:

Ribbon	Settings > Version Control > Package-VC or Publish > Model Exchange > Package Control > Package Control
Keyboard Shortcuts	Ctrl+Alt+P

Configure a Controlled Package

Option	Description
Control Package	Select the checkbox to indicate that this is a controlled Package.
For all packages, create placeholders for external references	Select to force all Native/XMI imports of this Package to exclude incoming relationships and instead create external references. This overrides the 'Create placeholders for missing External References during XMI 1.1/2.1 and Native File Import' checkbox in the XML Specifications options for a user, if that checkbox is not selected.
Version Control	Leave this field set to '(None)', to manage the Package under Native/XMI control. The 'Package Control Options' dialog is also used in adding a Package to Version Control and, in this context, is described under the <i>Version Control</i> section.
Filename	Type or browse for the path and Native/XMI file name into which to export the Package data and from which to import it. You can also enter Local Path Substitution strings; for example, an XMI local path definition where: <code>myLocalPath="C:\Documents and Settings\John\Desktop\EA Models"</code> So that %myLocalPath%\CIM.xml is equivalent to: <code>C:\Documents and Settings\John\Desktop\EA Models\CIM.xml</code>

Type	Click on the drop-down arrow and select the type of Native/XMI to be generated; options include: - Enterprise Architect XMI 1.1 - Enterprise Architect XMI 2.1 - Enterprise Architect Native XML - Enterprise Architect Native XEA Any of the above options can be selected for complete import/export round tripping of Packages.
Version ID	Type the version number you want to apply to the Package. This field is the same as the 'Version' field on the Package 'Properties' dialog; any change to either field updates the other.
Owner	Defaults to the name of the Package owner; if necessary, type or select a different name.
Use DTD	(Optional) Select the checkbox to use a Data Type Definition (DTD) for the import and export. This option is available for Enterprise Architect XMI 1.1 type.
Log Import/Export	(Optional) Select the checkbox to log import and export activity to a log file.
Batch Import	(Optional) Select the checkbox to mark the Package as a Batch Import Package.
Batch Export	(Optional) Select the checkbox to mark the Package as a Batch Export Package.
Include sub-packages	Select the checkbox to include the entire sub-Package hierarchy of this branch in the exported Native/XMI (default). (For an Enterprise Architect XMI 1.1 export) Deselect the checkbox to include only the immediate contents of the Package in an XMI export (XMI stubs).
OK	Click on this button to set the Package Control options and close the dialog.

Notes

- In the Corporate, Unified 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 XMI 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

Remove Package from Control

If required, you can remove the control from a Package. Before removing the Package control, and if it is being used for Version Control, you must check in the Package.

Access

Select a Package in the Browser window, then:

Ribbon	Settings > Version Control > Package-VC or Publish > Model Exchange > Package Control > Package Control
Keyboard Shortcuts	Ctrl+Alt+P

Remove control from a Package

Step	Action
1	Select the Package and the 'Package Control' ribbon 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

Save a Package

Once you have correctly configured a Package, you can save the controlled Package to the designated Native/XMI file.

Access

Select a Package in the Browser window, then:

Ribbon	Publish > Model Exchange > Package Control > Save to File
Context Menu	Right-click on Package > Package Control > Save Package to File
Keyboard Shortcuts	Ctrl+Alt+S

Save a controlled Package to file

Step	Action
1	Select the Package and the 'Save Package to File' ribbon 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

Load a Package

If a Package has been marked for control it is displayed in the Browser window with a red rectangle to the left of the Package icon. If you have previously saved the controlled Package, you can reload it.

Access

Select a Package in the Browser window, then:

Ribbon	Publish > Model Exchange > Package Control > Load from File
Context Menu	Right-click on Package > Package Control > Load Package from File
Keyboard Shortcuts	Ctrl+Alt+L

Load a controlled Package from an XMI file

Step	Action
1	Select the Package and the 'Load from file' ribbon option. If you have configured the Package Control details, the system 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

Batch Export Controlled Packages

You can export a group of Controlled Packages to Native/XMI format in one operation.

Access

Ribbon	Publish > Model Exchange > Export > Batch Export Controlled Packages
--------	--

Export a group of Controlled Packages

Step	Action
1	Open the 'Batch Export Controlled Packages' dialog.
2	The 'Batch Export Controlled Packages' dialog lists all of the Controlled Packages in the current Project. 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.
4	To save this configuration as the default, click on the Save Settings button.
5	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 Native/XMI file and proceeds to the next Package.

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 'Preferences' dialog
- The option is applicable only to XMI 1.1 Controlled Packages

Batch Import Controlled Packages

You can import a group of Controlled Packages from Native/XMI format into Enterprise Architect in one step.

As the system processes each Package, it updates the 'Status' column against each Package name on the 'Batch XMI Controlled Packages' 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

Ribbon	Publish > Model Exchange > Import Package > Batch Import Controlled Packages
--------	--

Import a group of Controlled Packages

Step	Action
1	Open the 'Batch XMI Controlled Packages' dialog using one of the access methods.
2	The 'Batch XMI Controlled Packages' dialog lists all of the Controlled Packages in the current Project. 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.

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

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.

Prepare for using XMI for Version Control

Step	Action
1	Select suitable Packages in the Browser window, to be marked as Controlled Packages.
2	Configure these with filenames that are visible to a Version Control system of your choice.
3	Save the Controlled Packages to establish a model base and check these into the Version Control system.

Apply Version Control

Step	Action
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.

Recover an earlier version

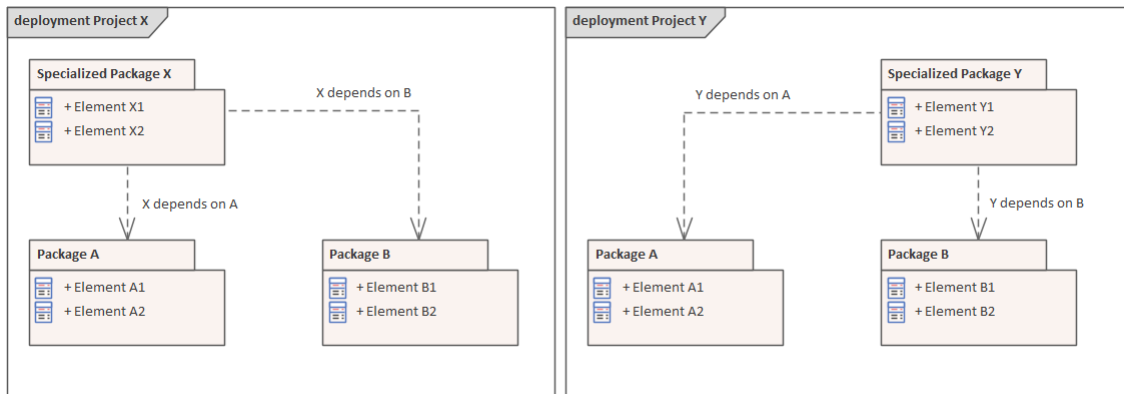
Step	Action
1	Save the current version first. This is important, because the Package is completely deleted during the import process. If necessary, manually update the Version Control system.
2	Get the required Package version from the Version Control system.
3	Select the Package to reload.

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.
---	---

Report Deletion of Cross Package References

Your model might contain at least one XMI 1.1 Controlled Package that is also used in other models, but not necessarily in the same context.

For example, consider the combination of Projects and Packages described in this figure:



Two common Packages, 'Package A' and 'Package B', are shared across two separate projects, 'Project X' and 'Project Y', through the use of XMI Package files. From the point of view of the common Packages, each of these projects is **'incomplete'**. Project X does not contain 'Specialized Package Y' and similarly, Project Y does not contain 'Specialized Package X'.

Problem

When 'Package B' is exported from 'Project X', its XMI file will **not** contain the cross-Package reference 'Y depends on B', as that relationship does not exist in 'Project X'. If the XMI file is subsequently imported into 'Project Y', the relationship 'Y depends on B' will be removed from 'Project Y'.

Solution

You can easily check for and preserve the cross-Package relationships by selecting the option 'Report Cross Package Reference Deletions (XMI 1.1)' in the 'Preferences' dialog ('Start > > Appearance > Preferences > Preferences' ribbon option, then select the 'XML Specifications' tab). When you save a controlled Package, Enterprise Architect compares the XMI for that save with the most recent XMI 1.1 revision of the Package. Any items that are missing from the Package are listed on the 'Missing Cross References' dialog.

Actions

Action	Detail
Decide which cross-references to merge	In some cases Enterprise Architect cannot determine whether a cross-reference is missing or intentionally deleted by the user. When the 'Missing Cross References' dialog displays, the basic analysis process should be: • If the item was deleted intentionally, deselect the checkbox beside it • If the item was not deleted intentionally, its presence in the list suggests it is a missing cross reference and you should leave the checkbox selected

Merge missing items	Items missing from the exported Package are listed in the 'Missing Cross References' dialog. To regenerate them into the XMI file that is being saved: • Select the checkbox against each item type • Click on the OK button Items of any type that you do not select are omitted from the saved XMI file.
Abort or cancel this XMI Export	If you do not want to proceed with the export (or Check-in or Put Latest) operation: • Click on the Cancel button
Review the details of an entry	To display the details of an entry on the 'Missing Cross References' dialog: • Double-click on the type name The details of each item are shown on the 'Missing <item type>' dialog. For any items where information is missing from the XMI file, the missing details are grayed-out. To find the item with the missing cross-reference in the Browser window, either: • Right-click on the entry in the list and select the 'Find in Project Browser' option, or • Click on the item and press Alt+G
Copy a value from a column on the dialog to the clipboard	To preserve a copy of an item from the dialog on the clipboard: • Click on the column value to highlight it; the selected value is shown in the bottom left corner of the dialog • Select the text in the bottom left corner and press Ctrl+C to copy that value to the clipboard

Notes

- If you are not sharing any common Packages between multiple projects, then detection of missing cross references is not necessary, and you should disable the feature by deselecting the 'Report Cross Package Reference Deletions (XMI 1.1)' checkbox.

CSV Import and Export

Within Enterprise Architect, you can import and export information about elements to and from comma-separated value (CSV) text files. This involves up to three possible actions.

Actions

Action	Detail
Set Up 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.
Import From CSV	Once you have defined a CSV import specification, you can read in major element characteristics from a CSV text file.
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 Specifications

To import and export element data to and from Enterprise Architect using CSV files, you must first set up one or more CSV file specifications. A file specification defines the:

- Fields from the spreadsheet in the order they are imported or exported
- Source or target filename (optional) and
- Delimiter between columns

Once you have defined one or more specifications, you can select one in the 'CSV Import/Export' dialog as the current specification to apply during an import or export action. The facility only imports and exports elements (within Packages) and their properties; items such as Class attributes cannot be imported or exported through this mechanism.

XMI import/export provides a solution to this limitation, as does use of the Automation Interface.

Access

Ribbon	Publish > Model Exchange > CSV > CSV Exchange Specification
--------	---

Set Up a CSV Specification

Field	Action
Specification Name	If creating a new specification, type in a unique specification name. If editing an existing specification, click on the drop-down arrow and select the specification name.
Delimiter	Click on the drop-down arrow and select 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 any instances of " in the field are doubled (that is, " becomes "").
Notes	Type in a brief description of the specification.
Default Filename	Type in the default file location and file name to be used when importing and exporting. Alternatively, click on the  button and browse for the location.
Default Direction	Click on the drop-down arrow and select the default action - 'Import' or 'Export'. Each specification can be used in either direction, but this field provides the default.
Default Types	Type in a comma-separated list to limit the element types being exported; for example: class,requirement,component,node,object. • If you specify element types, ONLY elements of those types are exported or imported; so, for 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	Select this checkbox to include fields generated by Enterprise Architect, to embed/reconstruct the Package hierarchy.
Available Fields	Click on the name of an available field, and double-click on it to add it to the Specification (it displays at the end of the 'File Specification' list). Press and hold Ctrl while you select several fields, but use the Add field button to transfer them all to the 'File Specification' list.
File Specification	Review the record fields already assigned, listed in the order in which they are plotted across the spreadsheet. If you want to change the sequence of fields, click on a field name and click on the  or  button to move the field name up or down the list.
Add Tagged Value Field	Click on this button to add a Tagged Value column to the CSV Specification, to import or export the value that an element has for a particular tag. A prompt displays to select one of two options: - Value - to import or export the content of the Tagged Value 'Value' field; a column heading with the prefix 'TagValue_' 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 'TagNotes_' is added to the Specification In each case, the 'Add Tag Value Field' dialog displays. Select either the: 'Global Tagged Values' radio button, or 'Tagged Values from the selected Element' radio button, click on the Other Element button and browse for the name of the element that contains the Tagged Value Click on the 'Tagged Value' field drop-down list and select the required Tagged Value, then click on the OK button.
Add Field	Click on this button to move all selected fields from the 'Available Fields' list to the 'File Specification' list.
Remove Field	Click on this button to move all selected fields from the 'File Specification' list back to the 'Available Fields' list.
New	Click on this button to clear all the dialog fields to create a new specification.
Save	Click on this button to save all changes to the currently selected specification.
Save As	Click on this button to save the current specification under a new name.
Delete	Click on this button to delete the current specification.
Close	Click on this button to close this dialog.

Notes

- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Export XMI', 'Import XMI' and 'Configure Resources' permissions to use the 'CSV Import/Export Specifications' option
- In 'Available Fields' and 'File Specification', the record fields 'Created Date' and 'Modified Date' are not set when imported from CSV

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

It is highly recommended that you do not change these fields if they have been automatically generated by Enterprise Architect's CSV exporter.

Inserted Fields

Field	Action
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.

External Data

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

CSV Export

It is possible to export information about elements from your project in Enterprise Architect to a CSV text file. In this operation you make use of 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 helps Enterprise Architect to identify and update existing elements, rather than create duplicates.

Access

Ribbon	Publish > Model Exchange > CSV > CSV Import/Export
--------	--

Export Data in CSV Format

Field	Action
Package	Confirm the name of the current selected Package.
Specification	Click on the drop-down arrow and select the name of the existing export specification to use.
Edit/New	Click on this button to edit the export specification or create a new one. The 'CSV Import/Export Specification' dialog displays.
File	Type in the name and path of the file to export to, or click on the  button and browse for the file.
Types	Type in the element types to export: leave blank for all, or enter a comma-separated list of types. 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 'Package' as an element type, otherwise no Packages are exported in which to preserve the hierarchy. If you do not specify any element types, all elements including Packages are exported and the hierarchy can be preserved.
Code Page	Displays the Code Page used when importing and exporting. This can be changed via the 'Source Code Engineering' page of the 'Manage Model Options' dialog.
Action	Click on the 'Export' radio button to export to file.
Export Column Names	Select this checkbox to include the column names of the fields in the export file.
Include Subfolders	Select this checkbox to include subfolders and their child elements in the export.
Print Results	Click on this button to print out the result list.

View File	Click on this button to view the resultant CSV file with the default Windows application for CSV files.
Run	Click on this button to execute the export.
Close	Click on this button to cancel the export and exit the dialog.

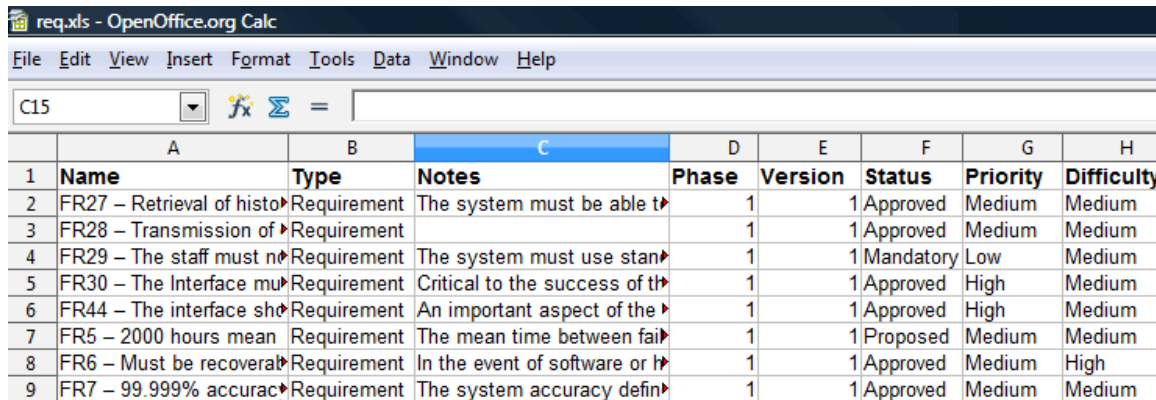
Notes

- In the Corporate, Unified 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

CSV Import

It is possible to import information about elements into your project in Enterprise Architect from a CSV text file. In this operation you make use of a CSV import 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 CSV source data file should resemble this:



The screenshot shows a spreadsheet window titled 'req.xls - OpenOffice.org Calc'. The spreadsheet contains a table with 9 rows and 9 columns. The columns are labeled A through H. The data is as follows:

	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 h	1	1	Approved	Medium	High
9	FR7 – 99.999% accurac	Requirement	The system accuracy defin	1	1	Approved	Medium	Medium

Note that the content of each CSV file cell must not be greater than 255 characters in length.

Access

Ribbon	Publish > Model Exchange > CSV > CSV Import/Export
--------	--

Import Data in CSV Format

Field	Action
Package	Confirm the name of the current selected Package.
Specification	Specify the name of the import specification to use.
Edit/New	Edit the import specification or create a new one.
File	Specify the spreadsheet filename to import from.
Types	Not used for import.
Code Page	Displays the Code Page used when importing and exporting. This can be changed via the 'Source Code Engineering' page of the 'Manage Model Options' dialog.
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.
Close	Click on this button to close 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, Unified 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

