UML Model Management

Enterprise Architect is an intuitive, flexible and powerful UML analysis and design tool for building robust and maintainable software.

This booklet explains the Model Management facilities of Enterprise Architect.

Enterprise Architect - UML Model Management

Printed: May 2010

Publisher

Sparx Systems

Managing Editor

Geoffrey Sparks

Technical Editors

Brad Maxwell
Simon McNeilly
Vimal Kumar
Henk Dekker

Special thanks to:

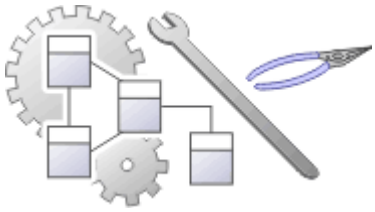
All the people who have contributed suggestions, examples, bug reports and assistance in the development of Enterprise Architect. The task of developing and maintaining this tool has been greatly enhanced by their contribution.

Table of Contents

Foreword

This user guide provides an introduction to the Model Management facilities of Enterprise Architect.

Model Management



What is a UML Model?

A *model* is a special type of package, being the top level entry point to an Enterprise Architect project file.

You can develop a project with one model, or with several. Each model is a *root node* of a hierarchy of Model Packages and Views and, below them, packages. A model contains the diagrams, elements, relationships and associated metadata that define the structure and function of a system or process. These components are organized through the package hierarchy, which helps to group and manage related components. By iterating through all models, you can access all the elements within the project.

You can create the model or models when you first create the project, or you can add and develop new models later. You can also delete a model, but remember that everything contained in the model is deleted as well.

In Model Management, you configure and maintain:

- Project files and data repositories 2
- Project sharing in team environments 3
- Version Control of packages 3
- User Security in updating packages and elements 3
- Auditing of model changes 3
- Baselining and differencing 3 to capture and roll back changes
- The Traceability 3 of model development
- Use of the Automation Interface 4 and Add-Ins 4 to automate and extend Enterprise Architect functionality
- The transfer of data 4 between projects in similar or different databases
- Replication 4 of models for remote sharing of development
- A Team Review 4 for distributing and discussing information concerning a model or project.

Project Files and Data Repositories

An Enterprise Architect project is stored in a data repository. Enterprise Architect enables you to work with .EAP files 6 (a Microsoft JET database). In the Enterprise Architect Corporate, Business and Software Engineering, System Engineering and Ultimate editions, you can also work with DBMS repositories 6 such as:

- SQL Server 40
- Access 2007 37
- MySQL 38
- Oracle 9i, 10g or 11g 43
- PostgreSQL 43
- Adaptive Server Anywhere 46
- MSDE Server 48
- Progress OpenEdge 48

Information on how to get started with projects can be found in the *Quick Start - Create a Project* topic in *Getting Started with Enterprise Architect*.

Project Sharing

Note:

This functionality is available in the Corporate, Professional, Business and Software Engineering, Systems Engineering and Ultimate editions. The Desktop edition is intended for single users, so does not support shared files.

Enterprise Architect enables project sharing 99 for efficient management of team development. You can create a replica of your project, make changes to it, then merge your changes back into the master project.

Version Control For UML Models

Enterprise Architect UML Model version control 130 enables you to:

- Coordinate sharing of packages between users, with either read-only access or update access
- Save and retrieve a history of changes to packages.

To use version control in Enterprise Architect, you require a third-party source-code control application such as *Subversion*, *CVS*, or any other version control product that complies with the Microsoft Common Source Code Control standard.

Version Control is documented in *Version Control Within UML Models Using Enterprise Architect*.

User Security

Note:

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

UML Model User Security 173 in Enterprise Architect provides a means of limiting access to update functions in a project. Elements can be locked per user or per group, and a password defined for login. Enterprise Architect offers two security policies:

- Standard, where each element is considered unlocked until specifically locked
- Rigorous, where each element is assumed to be locked until specifically unlocked.

User Security is documented in *User Security in UML Models*.

Traceability

Traceability 217 identifies the way a given process has been, or is to be, developed in a system. The process can be an internal, model-management process, where you monitor work by asking questions such as 'what work has been done to realize this Requirement or Use Case?', or a business or system process that is being modeled, where you ask questions such as 'what Requirements, Use Cases, Classes, Components, Test Cases and other elements define the implementation and deployment of this process?'

Audit UML Models

Auditing 197 is a project-level feature, available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions, that enables you to record model changes in Enterprise Architect. By enabling this option, model administrators can view a range of information regarding changes, such as:

- *Who* changed an element
- *How many* elements they changed
- *When* they changed the data
- *What* the previous values were, and
- *What type* of elements they changed.

Auditing is documented in *Auditing UML Models*.

Baselines and Differences

The Enterprise Architect Corporate, Business and Software Engineering, System Engineering and Ultimate editions provide a facility to 'Baseline 208' or snapshot a model branch in XMI format at a particular point in

time, and store it within the model in compressed format. More than one baseline can be stored against a single Enterprise Architect package. Using Baselines, you can compare packages at the current and earlier stages of development, using the Compare (Diff) 210 utility. The Compare utility is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect. It enables you to compare the current model with either an exported or a version-controlled Enterprise Architect XMI file on disk, as well as with a Baseline.

Baselines and Differencing are documented in *Baseline UML Models*.

Project Data Transfer

Note:

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

Enterprise Architect enables you to transfer project data 79 between project data repositories, row by row, table by table.

The Automation Interface

The Enterprise Architect Automation Interface provides a way of accessing the internals of Enterprise Architect models to, for example, perform repetitive tasks or produce custom reports. All development environments capable of generating ActiveX Com clients, such as Microsoft C# or Java, should be able to connect to the Automation Interface.

The Automation Interface is documented in the *Enterprise Architect Object Model* topic of the *SDK for Enterprise Architect*.

Add-Ins

Add-Ins are ActiveX COM objects that expose public Dispatch methods. The Enterprise Architect Add-In model builds on the features provided by the Automation Interface to enable you to extend the Enterprise Architect user interface and add functionality.

Add-Ins are documented in the *Enterprise Architect Add-In Model* topic of the *SDK for Enterprise Architect*.

Project Team Review

Enterprise Architect provides a Project Team Review, which can be used to discuss the development and progress of a project or model. You can switch the team review to other projects, so you can monitor and compare developments in several projects at once.

The Project Team Review is documented in *Using Enterprise Architect - UML Modeling Tool*.

Replication

Note:

This functionality is available in the Corporate, Professional, Business and Software Engineering, Systems Engineering and Ultimate editions. The Desktop edition is intended for single users, so does not support replication.

In addition to sharing projects in real time over a network, Enterprise Architect also enables projects to be shared using replication 100. Replication is a simple process that enables data interchange between .EAP based repositories and is suitable for use in situations where many different users work independently. Modelers merge their changes into a Design Master only as required. It is recommended that a backup is carried out prior to replication.

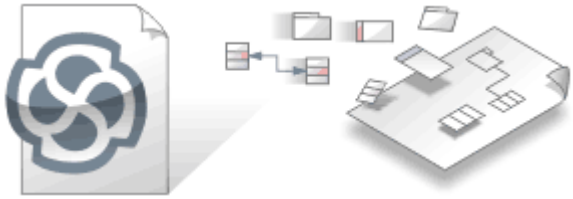
Replication requires the use of .EAP based repositories, and cannot be performed on repositories stored on a DBMS server.

See Also

- Upgrading Models 74
- Project Data Integrity 76
- Setting Up a Database Repository 11

- Model Maintenance 84
- Manage Views 85
- XMI Import and Export 104
- Team Development 98
- Spell Checking (see *Using Enterprise Architect - UML Modeling Tool*)
- Reference Data 226
- The WAN Optimizer 72

1 Enterprise Architect Project Files



An Enterprise Architect project 7 is stored in a data repository. In Enterprise Architect Desktop and Professional editions, you work with a single file having a .EAP extension. In Enterprise Architect Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions you can use a suitable DBMS database for project files.

Project Files

.EAP Files

In Enterprise Architect Desktop and Professional editions, a single file with a .EAP extension is used to store projects. A .EAP file is a Microsoft JET database, so you can also open it using MS Access 97, 2000 or 2003, or any other reporting tool that can work with JET databases.

DBMS Repositories

In Enterprise Architect Corporate Business and Software Engineering, Systems Engineering and Ultimate editions, you can use a suitable DBMS database for project files. DBMS project files have the same logical structure as .EAP files, but must be connected to using ADO/ODBC. See ***Connect to a Data Repository***, below.

Whenever you launch Enterprise Architect, the first thing displayed is the Start Page. From here, you can create a new project 9, open a project 8 and (Corporate, Business and Software Engineering, System Engineering and Ultimate editions) connect to a data repository 49.

Create a New Project File

On creating a new project 9, the Model Wizard 10 enables you to create a model containing various Model Packages.

You can also add Model Packages to a project from the Project Browser by:

- Right-clicking on an existing model and selecting the **New Model** or **Add a New Model using Wizard** context menu options
- Right-clicking on a package and selecting the **Add | Add a New Model using Wizard** context menu option
- Clicking on an existing model, pressing **[Insert]** and selecting the **New Model** or **Add a New Model using Wizard** context menu options
- Clicking on a package, pressing **[Insert]** and selecting the **Add a New Model using Wizard** context menu option.

Open an Existing Project

There are various ways to open a project 8 in Enterprise Architect. New users are advised to explore the EAExample file 8 supplied with Enterprise Architect.

Connect to a Data Repository

Note:

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

Enterprise Architect enables you to connect to any of the following data repositories:

- MS Access 97, 2000 and 2003 (in all editions - .EAP files are stored in Microsoft JET databases)
- Access 2007 37

- SQL Server 40 2000, 2005 and 2008
- MySQL 38
- Oracle 9i, 10g or 11g 43
- PostgreSQL 43
- MSDE 48
- Adaptive Server Anywhere 46
- Progress OpenEdge 68

To create a new data repository, you must first create a new database with the DBMS management software, then run supplied scripts to create the logical structure. You should then use Enterprise Architect data transfer functions to move a project from a .EAP or DBMS model into the new project.

1.1 What is a Project?

An Enterprise Architect project is a mechanism for storing and managing the components of one or more UML models.

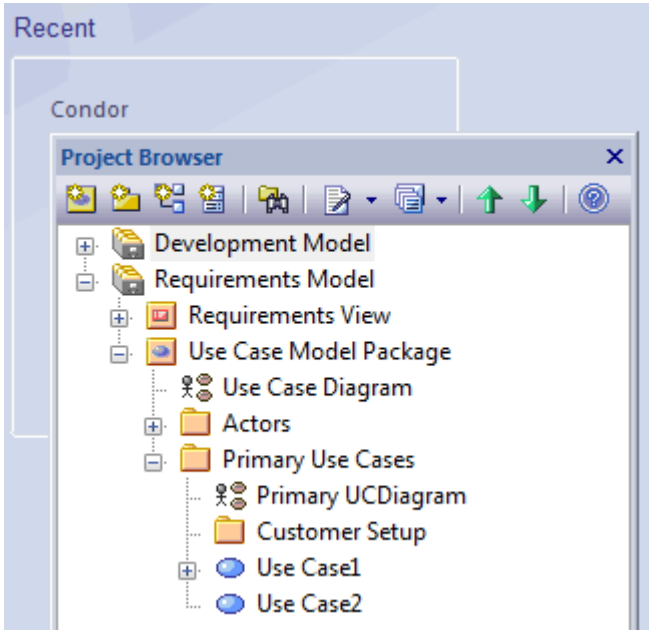
A project can be a .EAP file 6 in an MS Access database or (in the Enterprise Architect Corporate, Business and Software Engineering, System Engineering and Ultimate editions) a structure of files in a database management system 6 such as MySQL or Oracle.

A project can contain a single model, or a number of models, each of which defines a particular system or process. A model contains the diagrams, elements, relationships and associated metadata that define the structure and function of the system or process. These components are organized into a hierarchy of packages, which help to group and manage related components.

Different aspects of the process or system - or their development - are defined by Model Packages, which you generate 10 from templates specifically structured to support the aspects that the Model Packages represent, such as requirements or deployment (see the *Model Templates* topic in *Using Enterprise Architect - UML Modeling Tool*). You can generate these templated packages at any level of the hierarchy, but as they are created with their own content they are more useful at the top levels.

The top-level packages in a model can also be Views 85, which represent partitions of the model that you define yourself. You can start with standard Views such as Class or Component, or create whatever partitions are appropriate to your model.

So a typical project could have a structure something like the following:



The project *Condor* contains two models:

- *Development Model* and

- *Requirements Model*.

Requirements Model contains:

- *Requirements View* and
- *Use Case Model Package*.

Each View or Model Package contains packages. Use Case Model Package contains:

- *Actors* and
- *Primary Use Cases*.

It also contains the diagram *Use Case Diagram*, which could be an overview of the package structure or function. Each package itself can contain one or more diagrams, one or more packages, and several elements. The Primary Use Cases package contains the:

- *Primary UCDiagram*
- *Customer Setup* package
- *Use Case 1* element
- *Use Case 2* element.

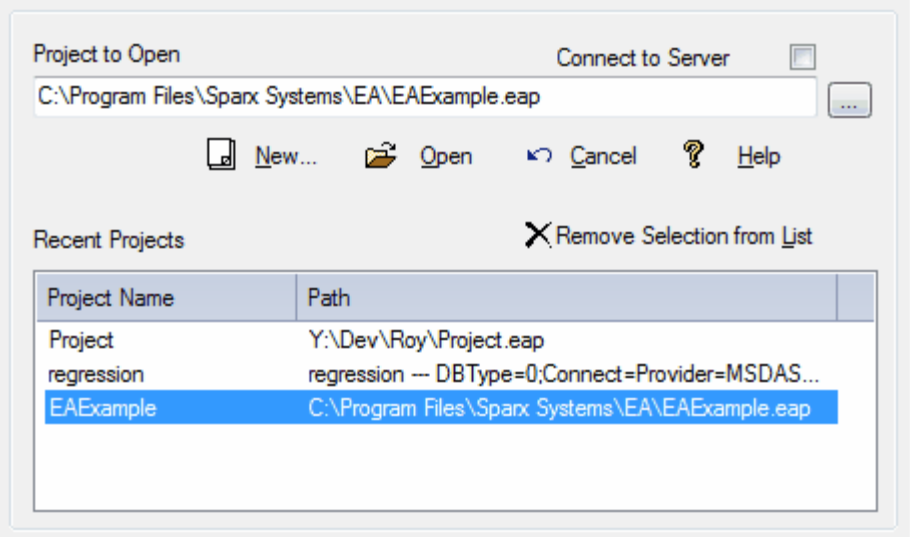
Each subordinate package also contains diagrams, elements and (if necessary) further packages. The elements are related by connectors created in the diagrams, and each element and connector has properties, attributes, operations and extensions defined in the respective Properties dialogs.

1.2 Open a Project

Enterprise Architect supports several different methods of opening an Enterprise Architect project file.

From the Main Menu

Select the **File | Open Project** menu option. From the Open Project dialog, select the path for the file to open and click on the **Open** button.



From the Default Tools Toolbar

(See *Using Enterprise Architect - UML Modeling Tool*)

Click on the folder icon to display the Open Project dialog, or the drop-down arrow to display a list of recently-opened projects.

From the Start Page

(See *Using Enterprise Architect - UML Modeling Tool*)

1. Click on **Open a Project File**. The Open Project dialog displays.

2. Use the file browser (**[...]**) to navigate to the project to open, which has a .EAP file extension (*.EAP). Select the project and click on the **Open** button.

Recently Opened Projects

Enterprise Architect keeps a list of recently opened projects and displays them on the Start Page for easy selection. If the project to open is in the Recent list, simply click once on the name of the project to open it.

Note:

If you already have a project open, Enterprise Architect prompts you to save changes before loading.

Enterprise Architect Example Project File

New Enterprise Architect users in particular should start by exploring the *EAExample* file supplied with Enterprise Architect. The example model file is stored in your Enterprise Architect installation directory. The default installation directories, depending on which version you have installed, are:

- Registered version: C:\Program Files\Sparx Systems\EA
- Trial version: C:\Program Files\Sparx Systems\EA Trial
- Lite version: C:\Program Files\Sparx Systems\EA Lite

Connect to a Data Repository 49

Note:

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

You also have the option to connect to SQL Server 40, MySQL 38, Oracle 9i, 10g or 11g 43, Postgre SQL 62, ASA 65, MSDE Server 68 and Progress OpenEdge 68 data repositories.

1.3 Create a New Project

Enterprise Architect projects can be created via the **File | New Project** menu option, which displays the Save New Enterprise Architect Project dialog. Select a directory and enter a file name for your project, then click on the **Save** button. Once the project has been saved, the Select Model(s) 10 dialog displays, which makes a selection of Model Packages available. Select the Model Packages to include and click on the **OK** button. Enterprise Architect adds a model containing the selected Model Packages to the Project Browser.

You can also add Model Packages to the project using the **New Model From Pattern** icon in the Project Browser toolbar. Alternatively, new projects can be created from the Start Page; select the **Create a New Project** option. (See *Using Enterprise Architect - UML Modeling Tool*).

The Model Wizard

The Model Wizard is used to add a selection of Model Packages to the project, through the Select Model(s) dialog.

The EABase Project File

The default project file (*EABase.EAP*) is supplied when you install Enterprise Architect. By default, the example project file is stored in your Enterprise Architect installation directory. The default installation directories, depending on which version you have installed, are:

- Registered version: C:\Program Files\Sparx Systems\EA
- Trial version: C:\Program Files\Sparx Systems\EA Trial
- Lite version: C:\Program Files\Sparx Systems\EA Lite

Design a Custom Template

You can customize any Enterprise Architect project and use it as the base for the new project. This enables you or your organization to build a default project with company standards, tutorials, frameworks or any other common piece of modeling already in-built. A default project is no different from an ordinary project; Enterprise Architect simply copies and renames it as a starter for your new project. With careful planning you can save

yourself many hours of work at project start-up.

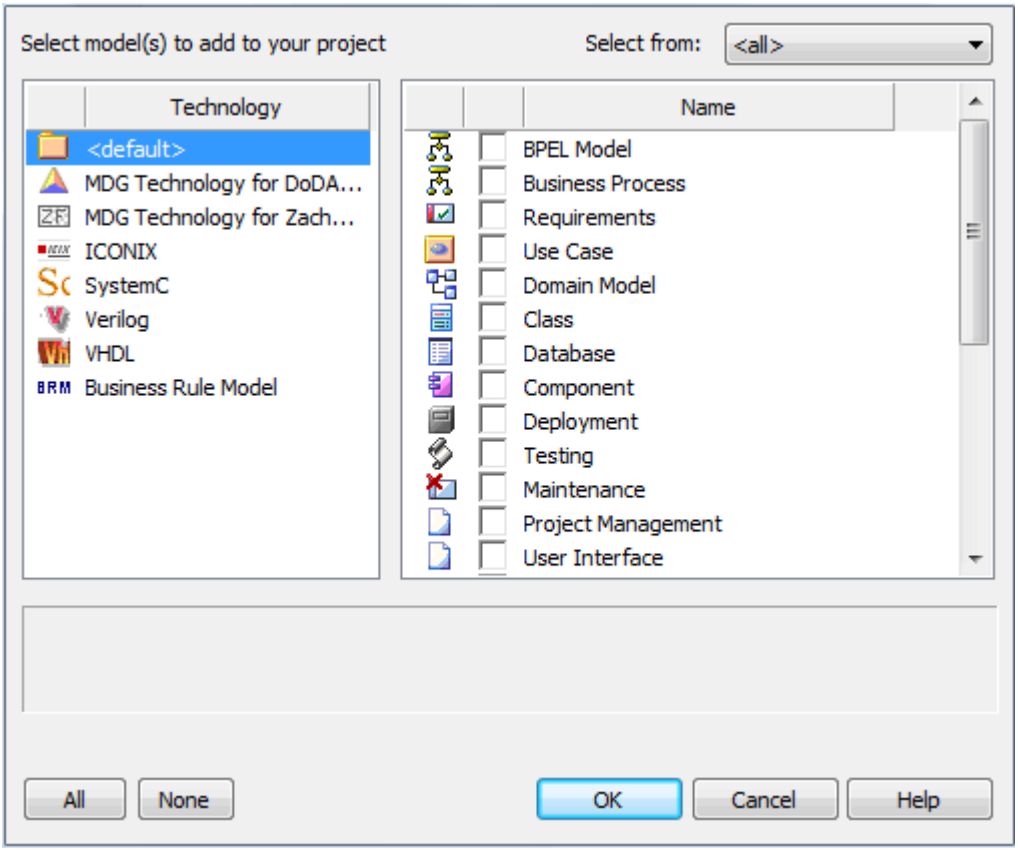
Configure Project

Having created your project, you can set a range of project parameters to define defaults, tailor the project to particular coding languages, and make development and use of the project consistent. See the following topics in *Using Enterprise Architect - UML Modeling Tool*:

- *The Settings Menu*
- *Defaults and User Settings*

1.3.1 Model Wizard

The Model Wizard is available on creating a new project. Once you have created a project, you can also access the Model Wizard from the Project Browser. Right-click on a root node, View or other top-level package and select the **Add a New Model using Wizard** context menu option. The Select Model(s) dialog displays.



Option	Use to
Select From	Select the category containing the types of Model Package to create.
Technology	(If you have advanced Add-Ins and MDG Technologies, this panel lists them). Select the appropriate technology to list the associated templates in the Name panel. To list the standard Enterprise Architect model templates, select **<default>**. For more information on the model templates, see *Using Enterprise Architect - UML Modeling Tool*.
All	Select all of the models.

Option	Use to
None	Clear all models selected.
OK	Create the Model Packages for your project.
Cancel	Abort the creation of model packages.
Help	Display this Help topic.

If you are a Technology Developer, you can also create and distribute custom templates as part of your own MDG Technology. The name of your technology displays in the Technology panel and, when you select the technology, the model template names display in the Name panel. If you have defined a filter in your Model Technology File, you can select that from the **Select from:** drop-down list (see *SDK for Enterprise Architect*).

1.4 Set Up a Database Repository

Introduction

The Desktop and Professional versions of Enterprise Architect use an MS JET database as the model repository.

If you purchase the Corporate, Business and Software Engineering, Systems Engineering or Ultimate edition, you can create and use any of the following data repositories:

- SQL Server 40 2000, 2005 or 2008
- MySQL 38 4 or 5
- PostgreSQL 43 7 or 8
- Adaptive Server Anywhere 8 or 9, or SQL Anywhere 10 or 11 46
- Access 2007 37
- Progress OpenEdge 48
- MSDE 48 or
- Oracle 9i, 10g or 11g 43.

You *upsize* the Enterprise Architect models (either existing or template) to use your selected DBMS. The process of upsizing a model is straightforward and comprises the following steps:

1. Install the DBMS software and create a database. Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic.
2. Run a script supplied by Sparx Systems (http://www.sparxsystems.com/resources/corporate/index.html#sql_scripts) to create the required tables.
3. Open Enterprise Architect and use the Project Data Transfer 79 function (select the **Tools | Data Management | Project Transfer** menu option) to move a model from a .EAP file to the DBMS repository.

Note:

You cannot move a model from a source .EAP file of a version earlier than 3.5.0.

Setting up a database repository is a two- or three-stage process: firstly, you set up an ODBC driver for your database; secondly, you create the repository tables using scripts downloaded from the Sparx Systems web site; and finally, you connect to the repository. Full instructions on all three stages are provided below.

Set Up an ODBC Driver

Setting up an ODBC driver is only necessary for *MySQL*, *PostgreSQL, Progress OpenEdge* and *Adaptive Server Anywhere*. These database management systems require specific set-up in order to operate as a repository. Other database management systems connect through OLE DB and do not require a driver, so this stage can be skipped.

Note:

For Oracle, the *Microsoft OLE DB Provider for Oracle* is not appropriate. You use the *Oracle Provider for OLE DB* 55 instead.

To find out how to set up an ODBC driver, go to:

- Set Up a MySQL ODBC Driver 25
- Set Up a PostgreSQL ODBC Driver 27
- Set Up an Adaptive Server Anywhere ODBC Driver 30
- Set Up a Progress OpenEdge ODBC Driver 35.

Create a Repository

To find out how to download the scripts and create the data repository tables, go to:

- Create a MySQL Data Repository 38
- Create an Access 2007 Repository 37
- Create a SQL Server Data Repository 40
- Create an Oracle Data Repository 43
- Create a PostgreSQL Data Repository 43
- Create an Adaptive Server Anywhere Data Repository 46
- Create an MSDE Server Data Repository 48
- Create a Progress OpenEdge Data Repository 48

Connect to a Repository

Once the repository is created, you can connect to it.

Note:

To connect to a repository, you must have the usual SELECT, UPDATE, INSERT and DELETE permissions.

To find out how to connect to your repository, see one of the following topics:

- Connect to a MySQL Data Repository 50
- Connect to a SQL Server Data Repository 52
- Connect to an Oracle Data Repository 55
- Connect to a PostgreSQL Data Repository 62
- Connect to an Adaptive Server Anywhere Data Repository 65
- Connect to an MSDE Server Data Repository 68
- Connect to a Progress OpenEdge Data Repository. 68

1.4.1 Upsize to Access 2007

Before you set up Enterprise Architect for use with Access 2007, it is recommended that you run the project integrity check tool 76 (select the **Tools | Data Management | Project Integrity Check** menu option) on the base project to upsize to Access 2007. This ensures the project data is 'clean' before uploading.

Note:

You cannot move a model from a source .EAP file of a version earlier than 3.5.0.

Upsizing your Database

You upsize your database in two stages, as follows:

Stage One: Create the Repository

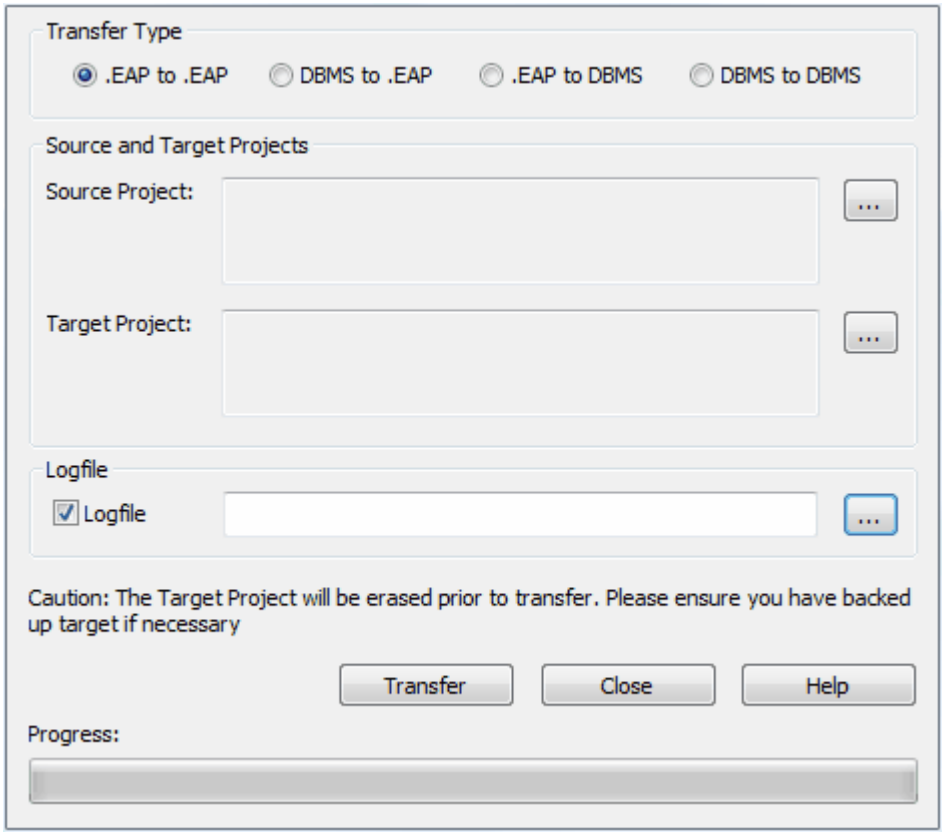
Using Access 2007, open a .EAP file and allow Access to convert it to a .ACCDB file. This forms the Access 2007 repository.

Note:

If you do not have Access 2007, you can connect by downloading the Access Database Engine from: http://www.microsoft.com/downloads/details.aspx?FamilyID=7554F536-8C28-4598-9B72-EF94E038C891&displaylang=en

Stage Two: Transfer the Data

1. Open Enterprise Architect. Click on the **Cancel** button on the Open Project screen to open with no project loaded.
2. Select the **Tools | Data Management | Project Transfer** menu option. The Project Transfer dialog displays:



3. In the Transfer Type panel, select .**EAP to DBMS**.
4. In the **Source Project** field, type the name of the .EAP file to upsize to Access 2007.
5. At the right of the **Target Project** field, click on the **[...]** (Browse) button. The Datalink Properties dialog displays.
6. Select **Microsoft Office 12.0 Access Database Engine OLE DB Provider** from the list, then click on the **Next** button.
7. On the Data Source Details page of the Connection dialog, type in the full path to the Access 2007 .ACCDB file.
8. Click on the **OK** button to return to the Project Transfer dialog.
9. If required, select the **Logfile** checkbox and type in a path and filename for the data transfer log file.
10. Click on the **Transfer** button to begin the data transfer process.

When the process is complete, you have upsized your model to Access 2007 and can now open it from Enterprise Architect.

1.4.2 Upsize to Sybase ASA

Before you set up Enterprise Architect for use with Sybase Adaptive Server Anywhere (ASA), it is recommend that you run the project integrity check tool (select the **Tools | Data Management | Project Integrity Check** 76 menu option) on the base project to upsize to ASA. This ensures the data is 'clean' before uploading.

Note:

You cannot move a model from a source .EAP file of a version earlier than 3.5.0.

Warning:

Before proceeding, ensure MDAC 2.6 or higher is installed on your system.

Upsizing Your Database

You upsize your database for ASA in three stages, as follows:

Stage One: Install ASA Components

1. Install Adaptive Server Anywhere - SQL Anywhere Studio 8 or higher. This also installs the ASA ODBC driver.
2. Create a new database for the Enterprise Architect repository using Sybase Central.
3. Create a suitable ODBC Data Source to point to your new database.

Note:

See Set up an Adaptive Server Anywhere ODBC Driver 30.

Stage Two: Configure the Database

From Sybase Central:

1. Right-click on the newly created database.
2. Open Interactive SQL and load the *ASA_BaseModel.sql* file. This is available to registered users on the Corporate edition Resources page of the Sparx website at http://www.sparxsystems.com/registered/reg_ea_corp_ed.html.
3. Run the script to create all required data structures.

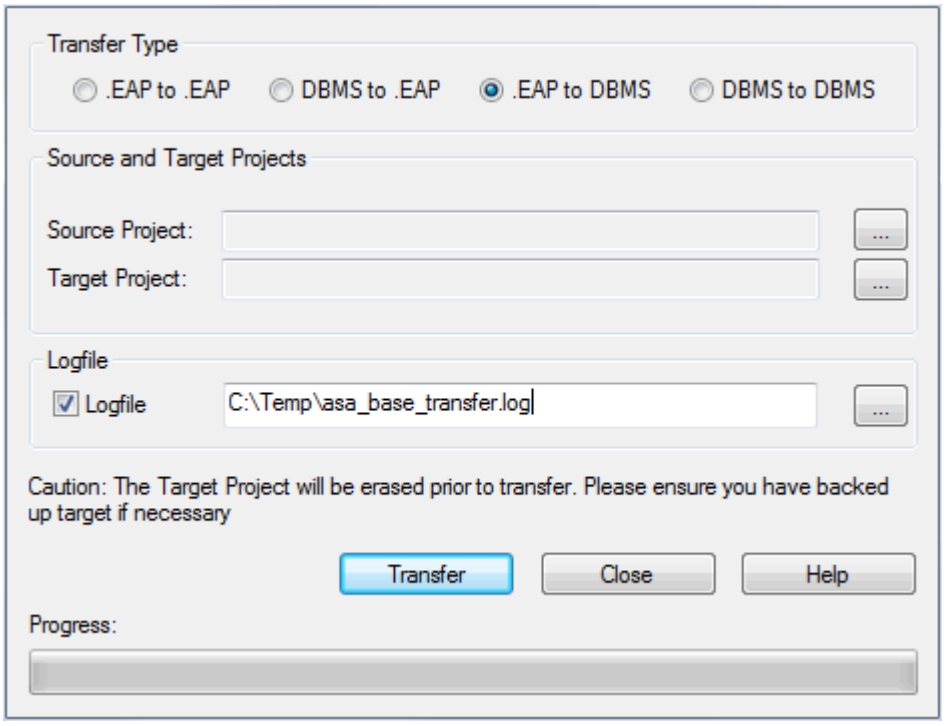
Note:

See Create a New Adaptive Server Anywhere Repository 46.

You now have an empty database, and can transfer an existing model into the server.

Stage Three: Transfer the Data

1. Open Enterprise Architect (click on the **Cancel** button on the Open Project screen to open with no project loaded).
2. Select the **Tools | Data Management | Project Transfer** menu option. The Project Transfer dialog displays.



3. In the Transfer Type panel, select .**EAP to DBMS**.
4. In the **Source Project** field, type the name of the .EAP file to upsize to ASA.
5. At the right of the **Target Project** field, click on the **[...]** (Browse) button. The Datalink Properties dialog displays.
6. Select **Microsoft OLE DB Provider for ODBC Drivers** from the list, then click on the **Next** button.
7. In the **Use Data source name** field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database.

Note:

See Connect to an Adaptive Server Anywhere Data Repository 65 for more information.

8. Click on the **OK** button.
9. If required, select the **Logfile** checkbox and enter a path for the data transfer log file.
10. Click on the **Transfer** button to begin the data transfer process.

When the process is complete, you have upsized your model to Adaptive Server Anywhere and can now open it from Enterprise Architect.

1.4.3 Upsize to Progress OpenEdge

Before you set up Enterprise Architect for use with OpenEdge, it is recommended that you run the project integrity check tool (select the **Tools | Data Management | Project Integrity Check** 76 menu option) on the base project to upsize to OpenEdge. This ensures the data is clean before uploading.

Note:

You cannot move a model from a source .EAP file of a version earlier than 3.5.0.

Warning:

Before proceeding, ensure MDAC 2.6 or higher is installed on your system.

Upsizing Your Database

You upsize your database for OpenEdge in three stages, as follows:

Stage One: Install OpenEdge Components

1. Install OpenEdge, version 10.0B3 or higher.
2. Install OpenEdge ODBC 10.0B or higher driver.
3. Create a suitable ODBC Data Source to point to your new database.

Note:

See Setup a Progress OpenEdge ODBC Driver 35.

Stage Two: Configure the Database

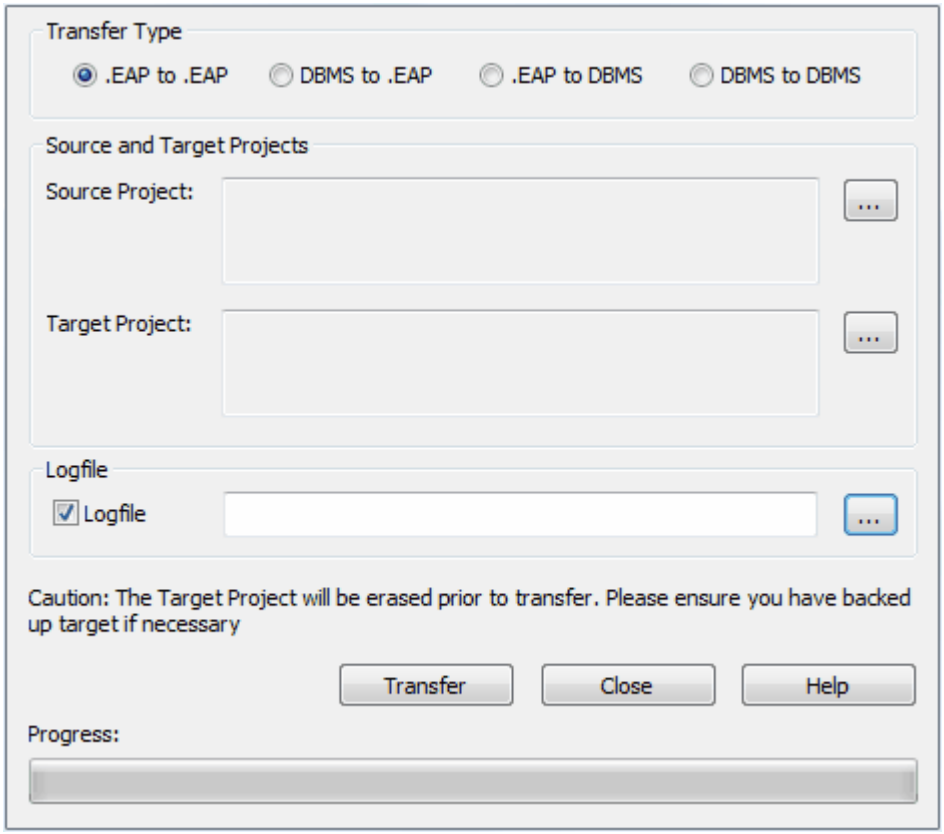
1. Create an empty OpenEdge database, using the scripts *OpenEdge_BaseModel.sql* file. This is available to registered users on the Corporate edition Resources page of the Sparx website at http://www.sparxsystems.com/registered/reg_ea_corp_ed.html.
2. Make sure the new database is selected as the current database.
3. Run the script to create all required data structures.

Note:

See Create a New OpenEdge Repository 48.

Stage Three: Transfer the Data

1. Open Enterprise Architect (click on the **Cancel** button on the Open Project screen to open with no project loaded).
2. Select the **Tools | Data Management | Project Transfer** menu option. The Project Transfer dialog displays:



3. In the Transfer Type panel, select .**EAP to DBMS**.

4. In the **Source Project** field, type the name of the .EAP file to upsize to OpenEdge.
5. At the right of the **Target Project** field, click on the [**...**] (Browse) button. The Datalink Properties dialog displays.
6. Select **Microsoft OLE DB Provider for ODBC Drivers** from the list, then click on the **Next** button.
7. In the **Use Data source name** field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database.

> **Note:**
>
> See Connect to a OpenEdge Data Repository 68 for more information.

8. Click on the **OK** button.
9. If required, select the **Logfile** checkbox and enter a path and filename for the data transfer log file.
10. Click on the **Transfer** button to begin the data transfer process.

When the process is complete, you have upsized your model to OpenEdge and can now open it from Enterprise Architect.

1.4.4 Upsize to MSDE

Before you set up Enterprise Architect for use with SQL Server Desktop Engine (MSDE), it is recommended that you run the project integrity check tool (select the **Tools | Data Management | Project Integrity Check** 76 menu option) on the base project to upsize to MSDE. This ensures the data is 'clean' before uploading.

> **Note:**
>
> You cannot move a model from a source .EAP file of a version earlier than 3.5.0.

> **Warning:**
>
> Before proceeding, ensure MDAC 2.6 or higher is installed on your system.

Upsizing Your Database

Follow the steps in Upsizing to SQL Server 20 to upsize your model to MSDE.

1.4.5 Upsize to PostgreSQL

Before you set up Enterprise Architect for use with PostgreSQL, it is recommend that you run the project integrity check tool (select the **Tools | Data Management | Project Integrity Check** 76 menu option) on the base project to upsize to PostgreSQL. This ensures your data is clean before uploading.

> **Note:**
>
> You cannot move a model from a source .EAP file of a version earlier than 3.5.0.

> **Warning:**
>
> Before proceeding, ensure MDAC 2.6 or higher is installed on your system.

Upsizing Your Database

You upsize your database for PostgreSQL in three stages, as follows:

Stage One: Install PostgreSQL Components

1. Install PostgreSQL, version 7.3.2 or higher.
2. Install psqlODBC, version 7.03.01.00 or higher (but do not use version 8.4.1).
3. Create a suitable ODBC Data Source to point to your new database.

Note:

See Set up a PostgreSQL ODBC Driver 27.

Stage Two: Configure the Database

1. From the PSQL command line, or using a tool such as the PostgreSQL command line, pgAdminIII or EMS PostgreSQL Manager, load the *Postgres_Basemodel.sql* file. This is available to registered users on the Corporate edition Resources page of the Sparx Systems website.
2. Run the script to create all required data structures.

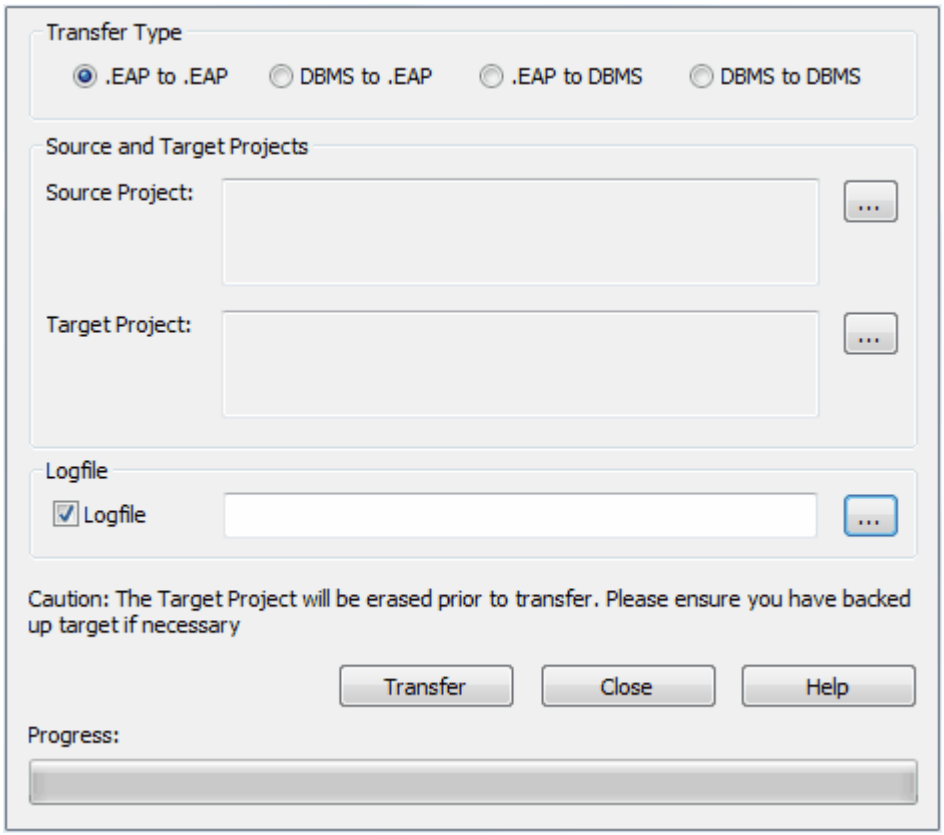
Note:

See Create a New PostgreSQL Repository 43.

You now have an empty database and can transfer an existing model into the server.

Stage Three: Transfer the Data

1. Open Enterprise Architect (click on the **Cancel** button on the Open Project screen to open with no project loaded).
2. Select the **Tools | Data Management | Project Transfer** menu option. The Project Transfer dialog displays.



3. In the Transfer Type panel, select **.EAP to DBMS**.
4. In the **Source Project** field, type or select the name of the .EAP file to upsize to PostgreSQL.
5. At the right of the **Target Project** field, click on the **[...]** (Browse) button. The Datalink Properties dialog displays.
6. Select **Microsoft OLE DB Provider for ODBC Drivers** from the list, then click on the **Next** button.
7. In the **Use data source name** field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database.

Note:

See Connect to a PostgreSQL Data Repository 62 for more information.

8. Click on the **OK** button.
9. If required, select the **Logfile** checkbox and type a path and filename for the data transfer log file.
10. Click on the **Transfer** button to begin the data transfer process.

Note:

If an error message displays reporting '*...nonstandard use of \\ in a string literal...*', set the server variable in the **postgresql.conf** file to:

```
escape_string_warning = off
```

When the process is complete, you have upsized your model to PostgreSQL and can now open it from Enterprise Architect.

1.4.6 Upsize to Oracle 9i, 10g or 11g

Before you set up Enterprise Architect for use with Oracle, it is recommended that you run the project integrity check tool (select the **Tools | Data Management | Project Integrity Check** 76 menu option) on the base project to upsize to Oracle. This ensures your data is clean before uploading.

Note:

You cannot move a model from a source .EAP file of a version earlier than 3.5.0.

Warning:

Before proceeding, ensure MDAC 2.6 or higher is installed on your system.

Upsizing Your Database

You upsize your database for Oracle in three stages, as follows:

Stage One: Create an Empty Database

1. Install Oracle.
2. Create an empty database.

Note:

See Create a New Oracle Repository 43.

Stage Two: Configure the Database

1. Using a tool such as the SQL*Plus or SQL Plus Worksheet, load the *Oracle_BaseModel.sql* file. This is available to registered users on the Corporate edition Resources page of the Sparx Systems website.
2. Make sure the new database is selected as the current database.
3. Run the script to create all required data structures.

Note:

See Create a New Oracle Repository 43.

You now have an empty database and can transfer an existing model into the server.

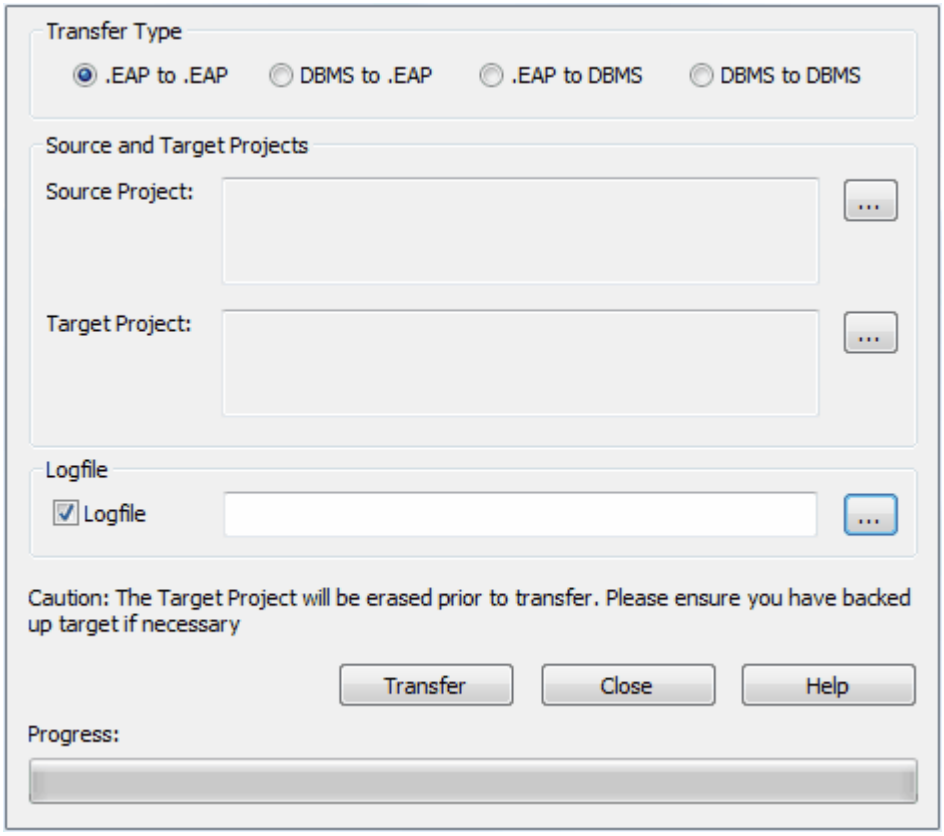
Stage Three: Transfer the Data

Note:

When transferring a project you must have permission to execute the **CREATE SEQUENCE** command.

1. Open Enterprise Architect (click on the **Cancel** button on the Open Project screen to open with no project loaded).

2. Select the **Tools | Data Management | Project Transfer** menu option. The Project Transfer dialog displays:



3. In the Transfer Type panel, select **.EAP to DBMS**.
4. In the **Source Project** field, type the name of the .EAP file to upsize to Oracle.
5. At the right of the **Target Project** field, click on the **[...]** (Browse) button. The Datalink Properties dialog displays.
6. Select **Oracle Provider for OLE DB** from the list, then click on the **Next** button.
7. On the Connection page of the Data Link Properties dialog, enter the Oracle service name in the **Data Source** field, and the user name and password as required.

Note:

See Connect to an Oracle Data Repository 55 for more information.

8. Click on the **OK** button.
9. If required, on the Project Transfer dialog, select the **Logfile** checkbox and type a path and file name for the data transfer log file.
10. Click on the **Transfer** button to begin the data transfer process.

Once the process is complete, you have upsized your model to Oracle and can now open it from Enterprise Architect.

1.4.7 Upsize to SQL Server

Before you set up Enterprise Architect for use with SQL Server, it is recommended that you run the project integrity check tool (select the **Tools | Data Management | Project Integrity Check** 76 menu option) on the base project to upsize to SQL Server. This ensure the project data is 'clean' before uploading.

Note:

You cannot move a model from a source .EAP file of a version earlier than 3.5.0.

Warning:

Before proceeding, ensure MDAC 2.6 or higher is installed on your system.

Upsizing Your Database

You upsize your database for SQL Server in three stages, as follows:

Stage One: Create an Empty Database

1. Install SQL Server.
2. Create an empty database.

Note:

See Create a New SQL Server Repository 40.

Stage Two: Configure the Database

1. Using a tool such as the SQL Query Analyser, load the *SQL Server - Base Model.sql* file. This is available to registered users on the Corporate edition Resources page of the Sparx Systems Website.
2. Make sure the new database is the currently active database.
3. Run the script to create all required data structures.

Note:

See Create a New SQL Server Repository 40.

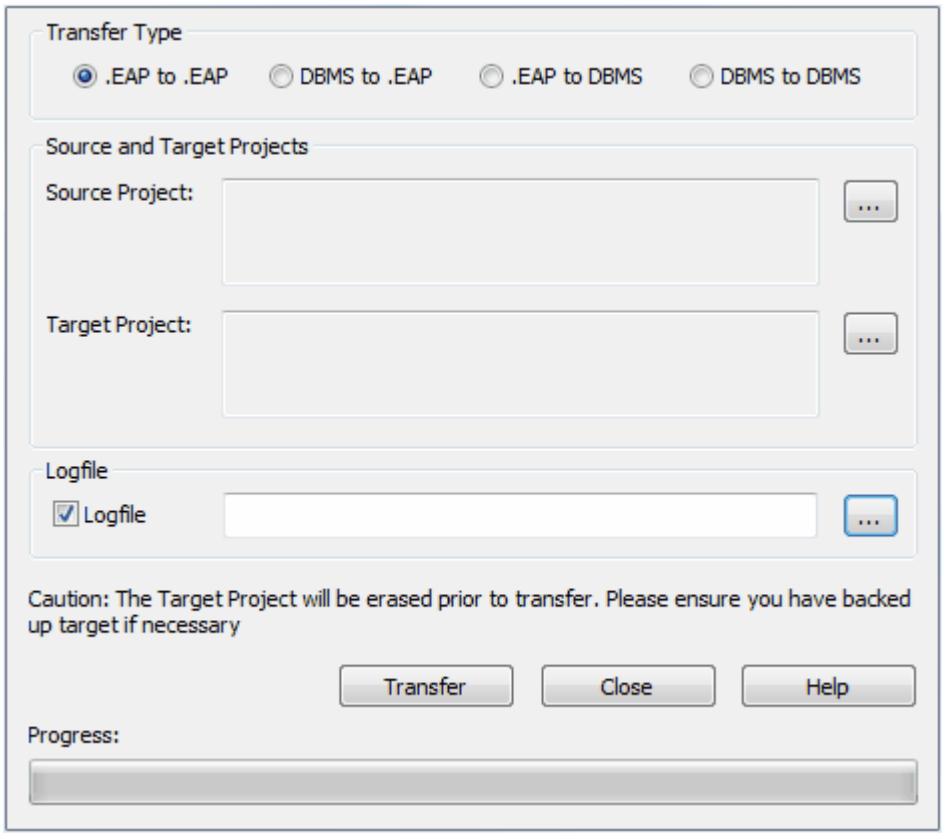
You now have an empty database, and can transfer an existing model into the server.

Stage Three: Transfer the Data

Note:

When transferring a project you must have *db_ddladmin* permission in order to execute the **SET IDENTITY_INSERT [table] {ON | OFF}** command.

1. Open Enterprise Architect (click on the **Cancel** button on the Open Project screen to open with no project loaded).
2. Select the **Tools | Data Management | Project Transfer** menu option. The Project Transfer dialog displays.



3. In the Transfer Type panel, select **.EAP to DBMS**.
4. In the **Source Project** field, type the name of the .EAP file to upsize to SQL Server.
5. At the right of the **Target Project** field, click on the **[...]** (Browse) button. The Datalink Properties dialog displays.
6. Select **Microsoft OLE DB Provider for SQL Server** from the list, then click on the **Next** button.
7. On the Data Source Details page of the Connection dialog, type in the server name, database name and security details as required.

Note:

See Connect to a SQL Server Data Repository 52 for more information.

8. Click on the **OK** button.
9. If required, select the **Logfile** checkbox and type in a path and filename for the data transfer log file.
10. Click on the **Transfer** button to begin the data transfer process.

When the process is complete, you have upsized your model to SQL Server and can now open it from Enterprise Architect.

1.4.8 Upsize to MySQL

Before you set up Enterprise Architect for use with MySQL, it is recommended that you run the project integrity check tool (the **Tools | Data Management | Project Integrity Check** 76 menu option) on the base project to upsize to MySQL. This ensures data is 'clean' before uploading.

Note:

You cannot move a model from a source .EAP file of a version earlier than 3.5.0.

Warning:

Before proceeding, ensure MDAC 2.6 or higher is installed on your system.

Upsizing Your Database

You upsize your database for MySQL in four stages, as follows:

Stage One: Install MySQL Components

1. Install MySQL version 4.0.3 or higher.
2. Install MySQL ODBC 3.51 or higher.
3. Create a suitable ODBC Data Source to point to your new database.

Note:

There are two critical non-default settings required; see Set up a MySQL ODBC Driver 25 and ensure you select the checkboxes in step 7.

Stage Two: Select Table Type

1. If you are using *InnoDB* tables, set up the *MySQL .ini* file as required and run the *MySQL - InnoDB BaseModel* script.
2. If you are using *MyISAM* tables, set up the *MySQL .ini* file as required and run the *MySQL - MyISAM BaseModel* script.

Note:

If MyISAM table types are used (default), transactional support is disabled. To enable transactions you must set up MySQL to use InnoDB tables and create the database tables as InnoDB type. Sparx provide a suitable script to create InnoDB based repository tables, as well as the more common MyISAM. These are available to registered users on the Corporate edition Resources page of the Sparx website at www.sparxsystems.com/registered/reg_ea_corp_ed.html.

Stage Three: Create the Database

1. Create an empty database.

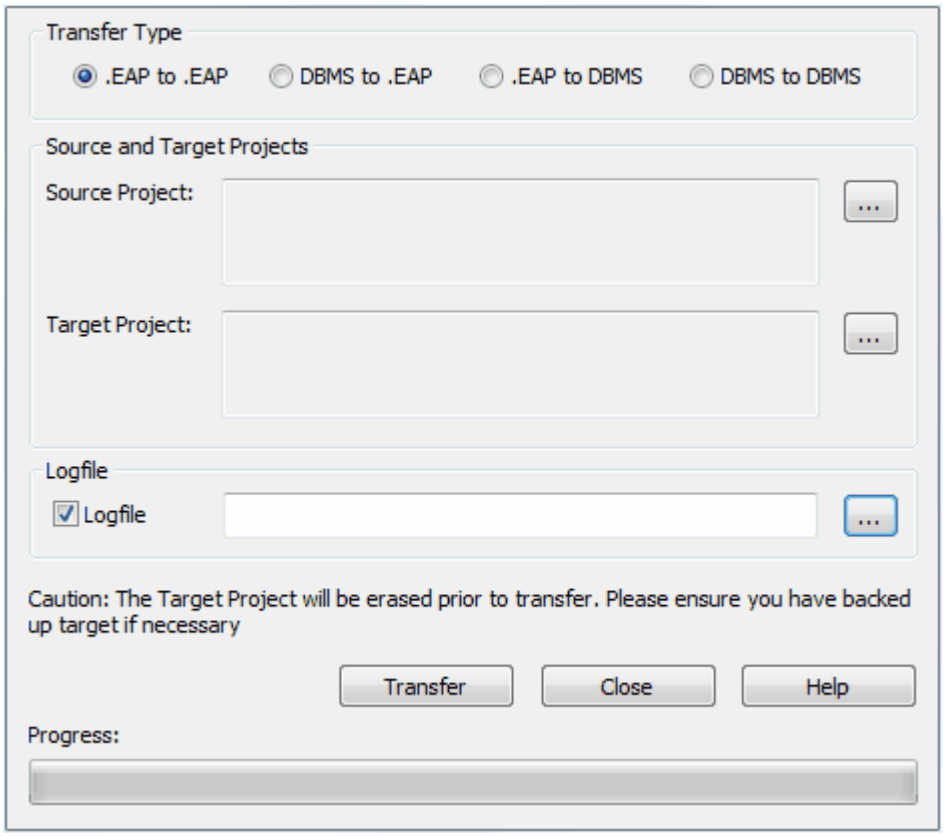
Note:

See Create a New MySQL Repository 38.

You now have an empty database, and can transfer an existing model into the server as described below.

Stage Four: Transfer the Data

1. Open Enterprise Architect (click on the **Cancel** button on the Open Project screen to open with no project loaded).
2. Select the **Tools | Data Management | Project Transfer** menu option. The Project Transfer dialog displays:



3. In the Transfer Type panel, select **.EAP to DBMS**.
4. In the **Source Project** field, type the name of the .EAP file to upsize to MySQL.
5. At the right of the **Target Project** field, click on the [**...**] (Browse) button. The Datalink Properties dialog displays.
6. Select **Microsoft OLE DB Provider for ODBC Drivers** from the list, then click on the **Next** button.
7. In the **Use Data source name** field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database.

Note:

See Connect to a MySQL Data Repository 50 for more information.

8. Click on the **OK** button.
9. If required, select the **Logfile** checkbox and type a path and filename for the data transfer log file.
10. Click on the **Transfer** button to begin the data transfer process.

When the process is complete, you have upsized your model to MySQL and can now open it from Enterprise Architect.

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

This topic details how to set up the following ODBC drivers, to enable connection to an Enterprise Architect data repository:

- MySQL ODBC Driver 25
- PostgreSQL ODBC Driver 27
- Adaptive Server Anywhere ODBC Driver 30
- Progress OpenEdge ODBC Driver 35

MySQL, PostgreSQL, Sybase Adaptive Server Anywhere and Progress OpenEdge require specific set-up in order to operate as a repository. Other database management systems connect through OLE DB and do not require a driver.

1.4.9.1 MySQL ODBC Driver

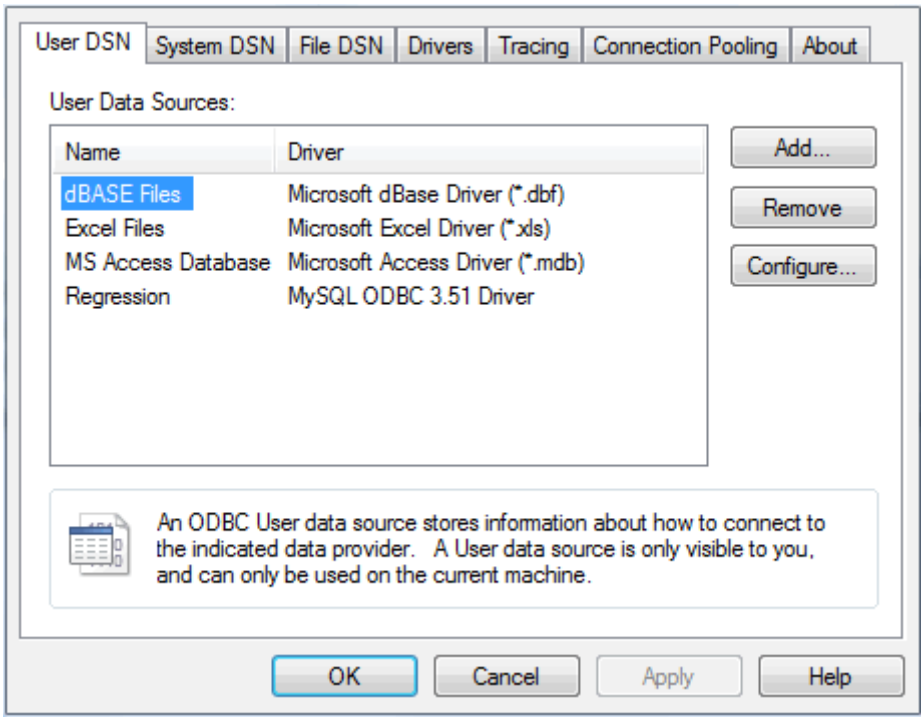
Before you can connect to a MySQL data repository, you must first set up a MySQL ODBC driver for which, in turn, you must first have installed Microsoft MDAC components, a MySQL DBMS system and a MySQL ODBC driver.

Note:

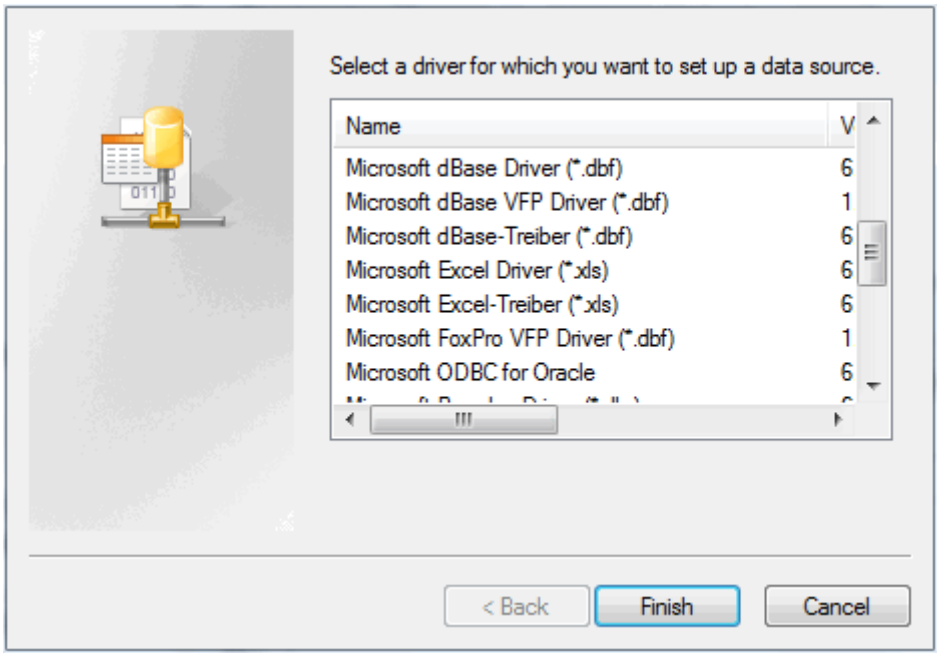
The recommended MySQL ODBC driver version is 5.1.5. (Version 3.51.14 creates problems in incorporating tests in elements.)

To set up your MySQL ODBC Driver, follow the steps below:

1. Select the Windows™ **Control Panel | Administrative Tools | Data Sources (ODBC)** option. The ODBC Data Source Administrator window displays.

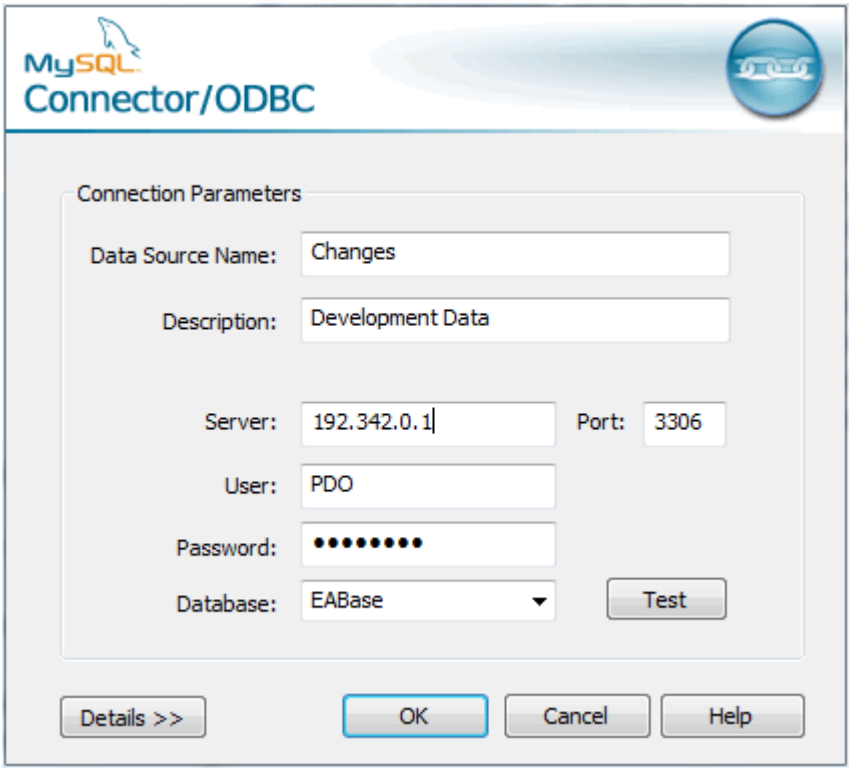


2. Click on the **Add** button. The Create New Data Source dialog displays, enabling you to add a new DSN.

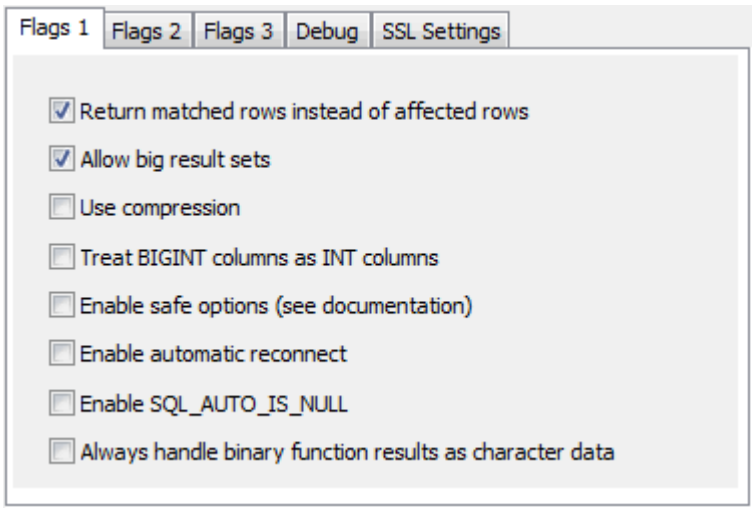


3. Select **MySQL ODBC 5.1 Driver** from the list.
4. Click on the **Finish** button. The MySQL Connector/ODBC dialog displays.
5. Enter the following configuration details:
   - A data source name for the connection
   - A description (optional)
   - The host address of the DBMS server
   - User name and password
   - The database name on the selected server.

   See the example below:



6. Click on the **Details>>** button and Flags 1 tab to set the advanced options.



7. Select the **Return matched rows instead of affected rows** and **Allow big result sets** checkboxes.
8. Click on the **Test** button to confirm that the details are correct.
9. If the test succeeds, click on the **OK** button to complete the configuration.
10. If the test does not succeed, review your settings.

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

1.4.9.2 PostgreSQL ODBC Driver

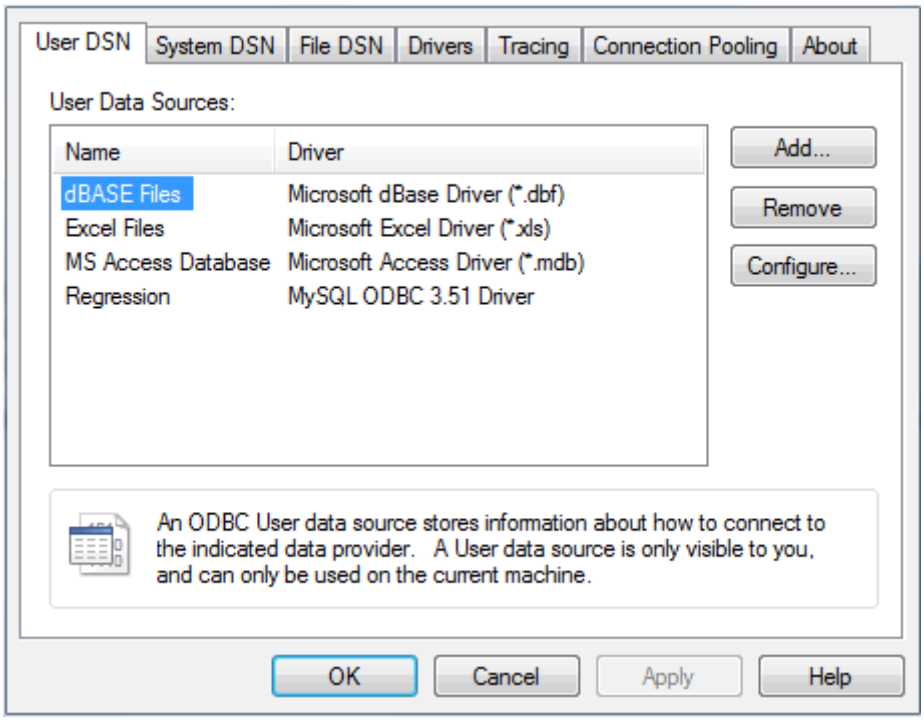
Before you can connect to a PostgreSQL data repository, you must first set up a PostgreSQL ODBC driver. To do this, you must have Microsoft MDAC components, a PostgreSQL DBMS system and a PostgreSQL ODBC driver (version 7.03.01.00 minimum) installed.

Note:

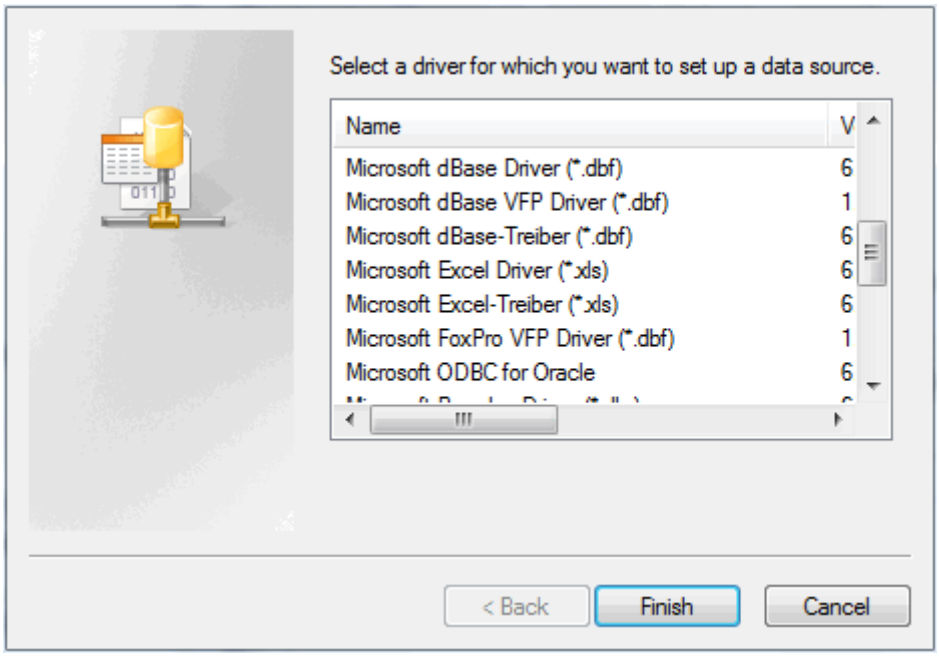
Do not use version 8.4.1 of the PostgreSQL ODBC Driver; it presents difficulties in transferring a project to a PostgreSQL repository.

To set up your PostgreSQL ODBC driver, follow the steps below:

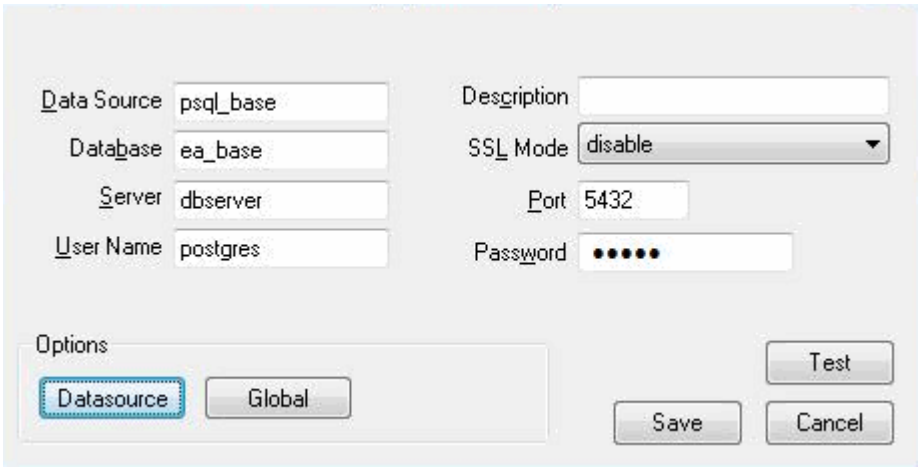
1. Select the Windows™ **Control Panel | Administrative Tools | Data Sources (ODBC)** option. The ODBC Data Sources Administrator window displays.



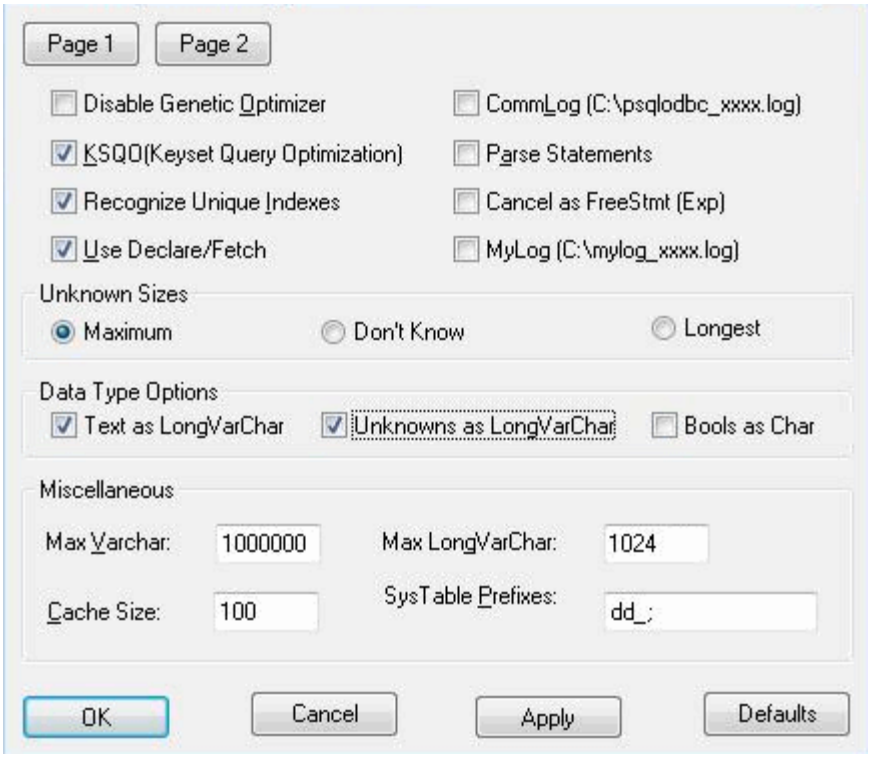
2. Click on the **Add** button. The Create New Data Source dialog displays, enabling you to add a new DSN.

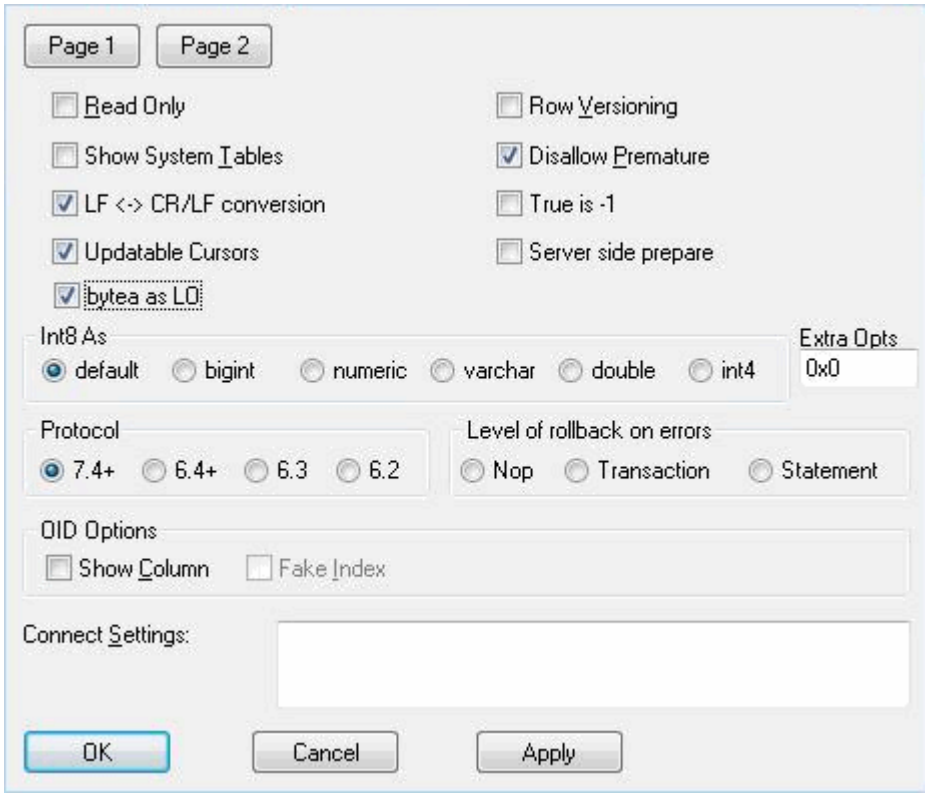


3. Select **PostgreSQL** from the list.
4. Click on the **Finish** button.
5. Enter the following configuration details:
   - A name for the connection
   - The actual name of the database.
   - Description (optional)
   - The host address of the PostgreSQL server.
   - User name and password.



6. Click on the **Datasource** button and set the options on Page 1 and Page 2 as shown on the examples below:





Note:

On Page 2, For PostgreSQL version 8+ select the **Disallow Premature** checkbox and, in the Protocol panel, select the **7.4+** radio button.

7. Click on the **OK** button to complete the configuration.

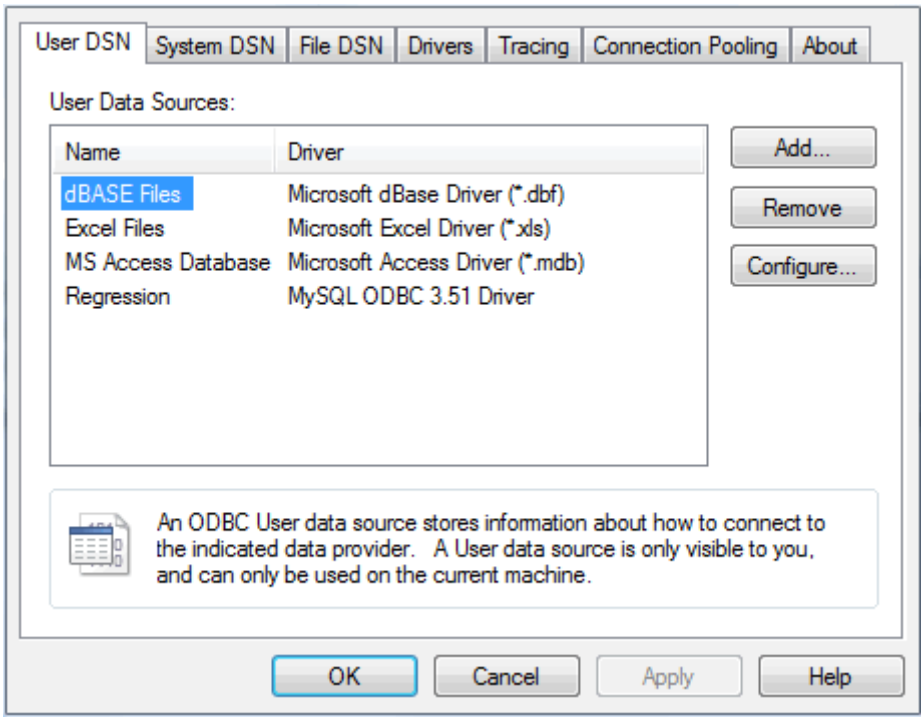
Your PostgreSQL connection is now available to use in Enterprise Architect.

1.4.9.3 ASA ODBC Driver

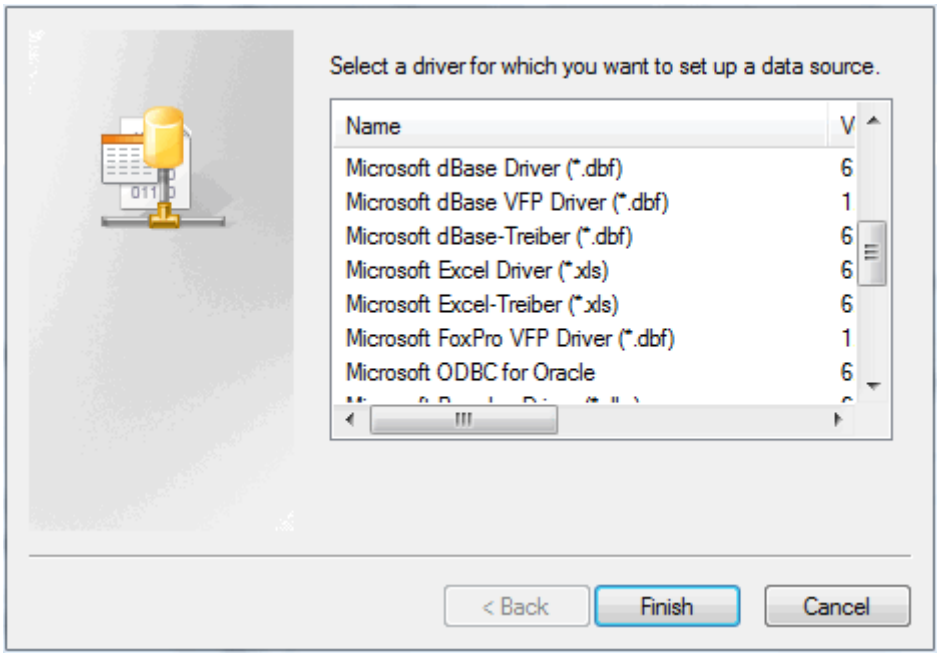
Before you can connect to an Adaptive Server Anywhere (ASA) data repository, you must first set up an ASA ODBC driver. To do this, you must have Microsoft MDAC components, the ASA DBMS system and the ASA ODBC driver (installed with the ASA DBMS) installed.

To set up your ASA ODBC Driver, follow the steps below:

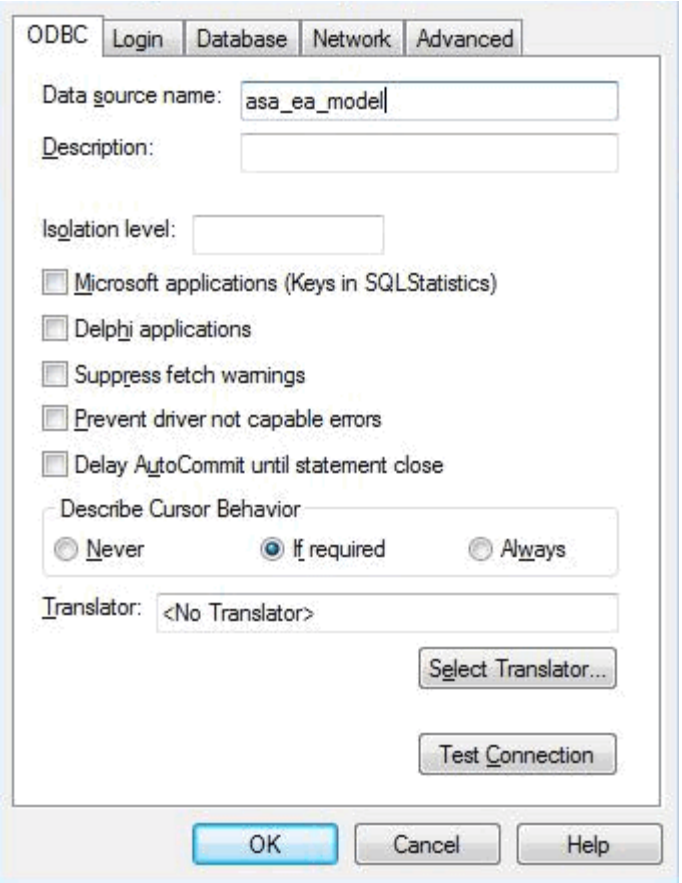
1. Select the Windows™ **Control Panel | Administrative Tools | Data Sources (ODBC)** option. The ODBC Data Sources Administrator window displays.



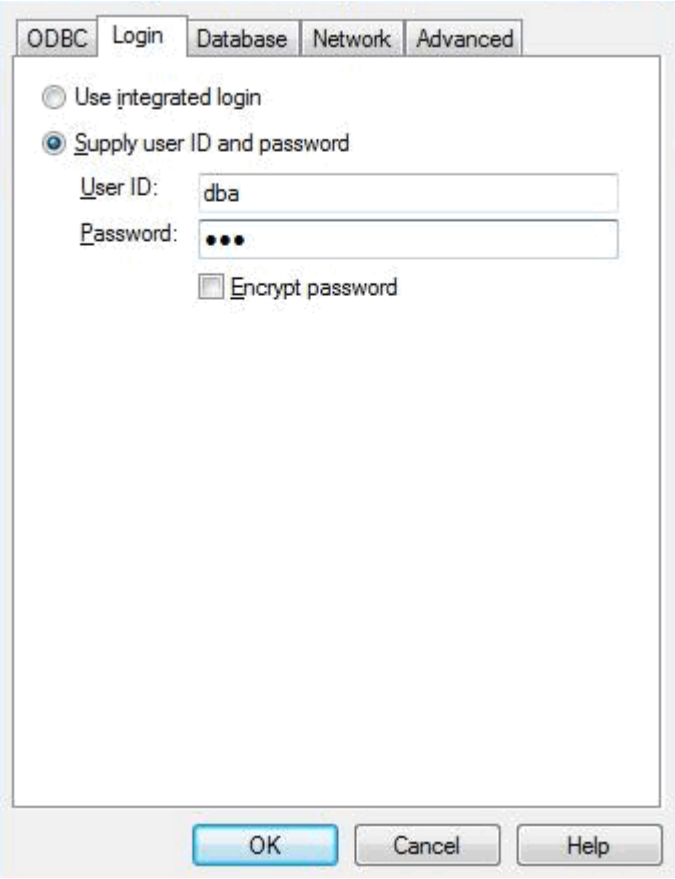
2. Click on the **Add** button. The Create New Data Source dialog displays, enabling you to add a new DSN.



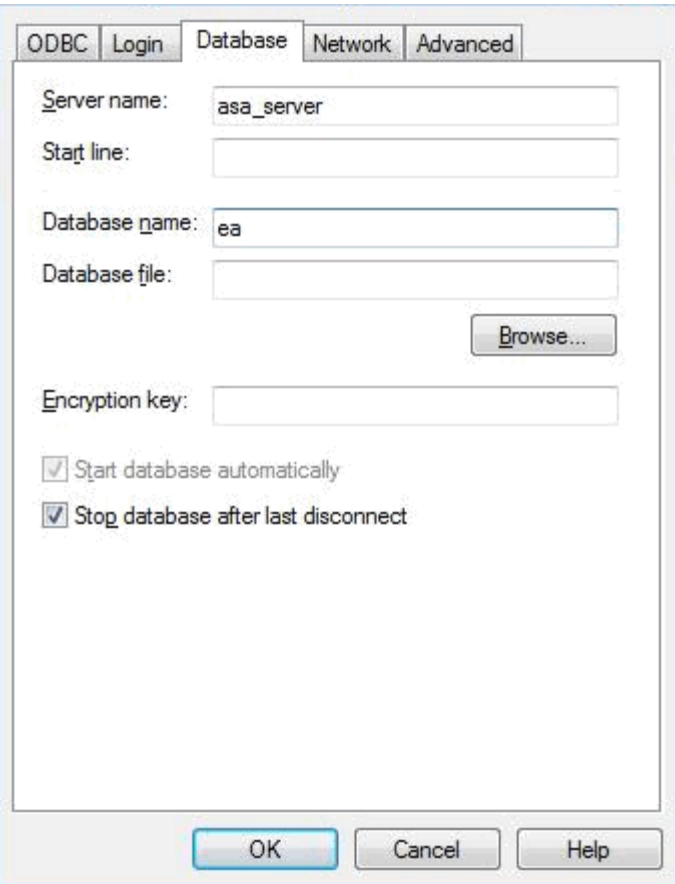
3. Select **Adaptive Server Anywhere** or **SQL Anywhere** from the list.
4. Click on the **Finish** button.
5. Enter the following configuration details:
   - A name for the connection on the ODBC tab.



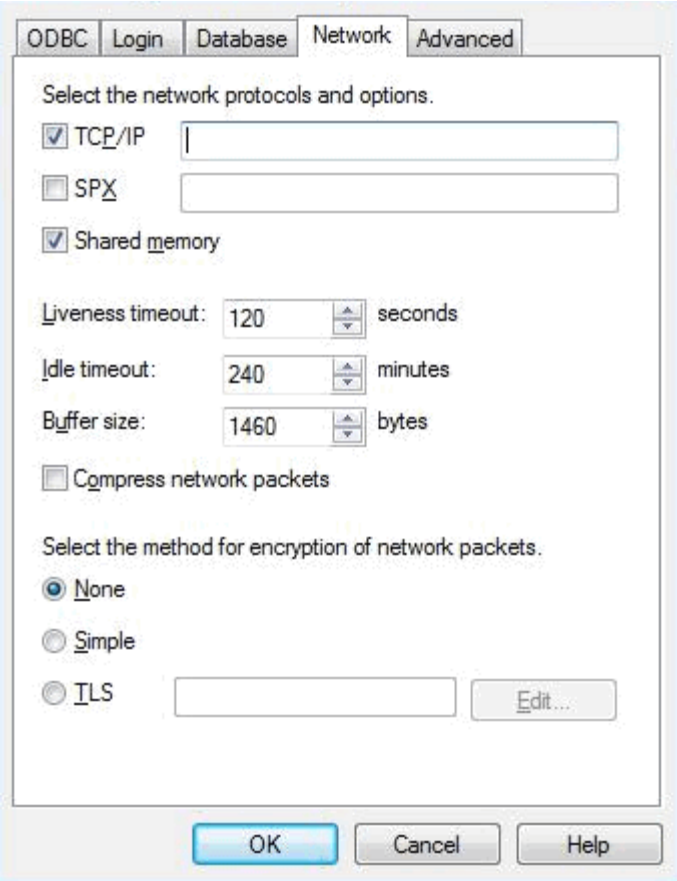
- The username and password on the Login tab (**dba**, **sql** are the defaults when ASA is installed).



- The server name and the path to the database, on the Database tab.



- The network protocol on the Network tab (if the database is on a network location).



6. You can now return to the ODBC tab and test the connection.
7. Click on the **OK** button to complete the configuration.

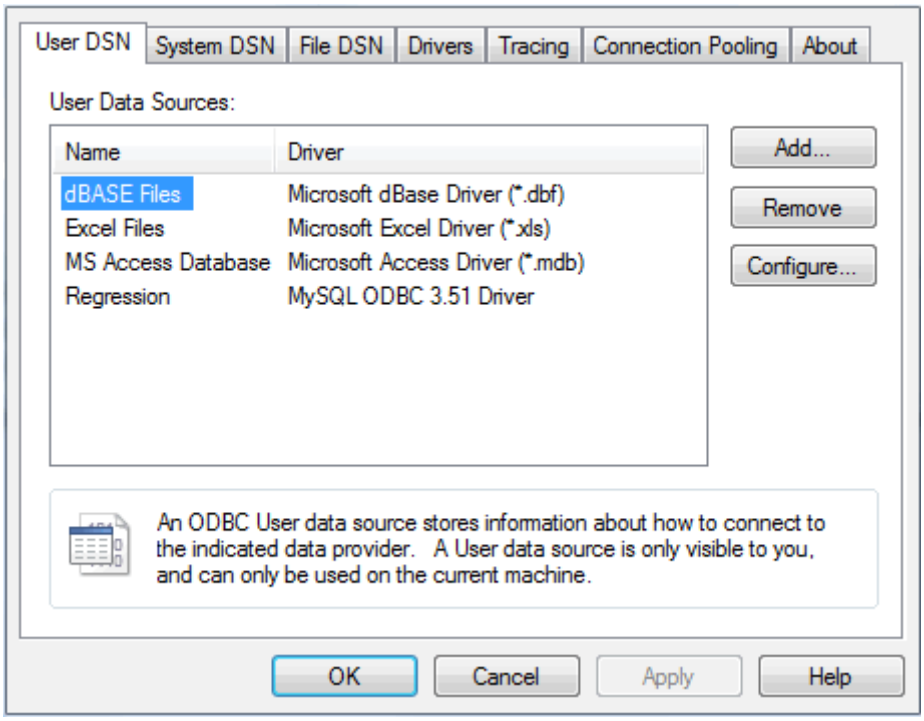
Your Adaptive Server Anywhere connection is now available to use in Enterprise Architect.

1.4.9.4 Progress OpenEdge ODBC Driver

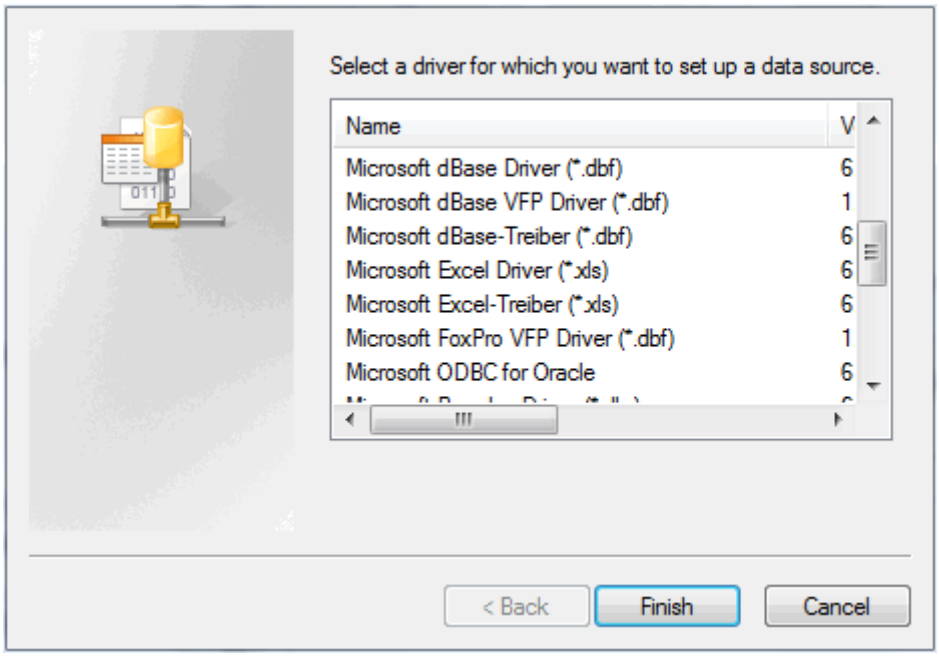
Before you can connect to an OpenEdge data repository, you must first set up an OpenEdge ODBC driver. To do this, you must have Microsoft MDAC components, OpenEdge DBMS system and DataDirect ODBC driver for OpenEdge (version 4.20 minimum) installed.

To set up the ODBC Driver, follow the steps below:

1. Select the Windows™ **Control Panel | Administrative Tools | Data Sources (ODBC)** option. The ODBC Data Sources Administrator window displays.



2. Click on the **Add** button. The Create New Data Source dialog displays, enabling you to add a new DSN.



3. Select the **DataDirect/OpenEdge SQL** driver from the list.
4. Click on the **Finish** button. The DSN Configuration dialog displays.
5. Enter the following configuration details:
   - The **Data Source Name**
   - The **Description** (optional)
   - The **Host Name** and **Port Number** of the DBMS server
   - The **Database Name** on the selected server
   - The **User ID**.

See the example below:

6. Click on the **Test Connect** button to confirm that the details are correct.
7. If the test succeeds, click on the **OK** button to complete the configuration.
8. If the test does not succeed, review your settings.

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

1.4.10 Create a Repository

This topic details how to create the following data repositories:

- Access 2007 37
- MySQL 38
- SQL Server 40
- Oracle 9i, 10g or 11g 43
- PostgreSQL 43
- Adaptive Server Anywhere (ASA) 46
- MSDE Server 48.

1.4.10.1 Access 2007

Note:

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

Using Access 2007, open a .EAP file and allow Access to convert it to a .ACCDB file. This forms the Access

2007 repository.

1.4.10.2 MySQL Repository

Note:

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

Before creating a MySQL data repository in Enterprise Architect, you must set up the MySQL and MySQL ODBC drivers. For further information on setting these up, see MySQL ODBC Driver 25.

To create a new MySQL repository, you must first create a database into which to import the table definitions for Enterprise Architect. Sparx Systems provide SQL scripts to create the required tables; how you create the database and execute that script are up to you.

- Registered users can obtain the scripts from the Registered Corporate edition Resources page of the Sparx Systems website at http://www.sparxsystems.com/registered/reg_ea_corp_ed.html
- Trial users can obtain the scripts from the Corporate edition Resources page of the Sparx Systems website at http://www.sparxsystems.com/resources/corporate/.

Create the Data Repository

Once you have created the database and executed the script, you should have an empty Enterprise Architect project to begin working with. You can transfer data from an existing .EAP file or simply start from scratch.

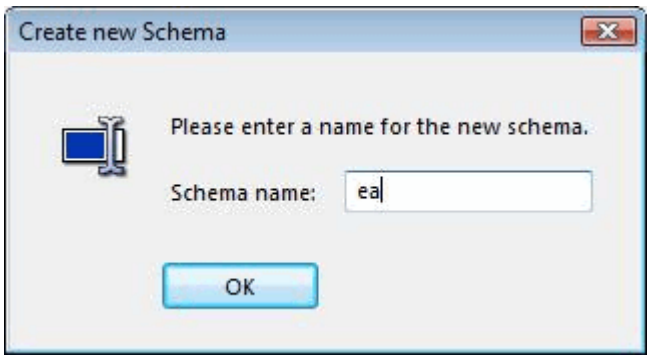
Third Party Tools

If you are unfamiliar with MySQL and DBMS systems in general, you might want to consider a suitable front end tool.

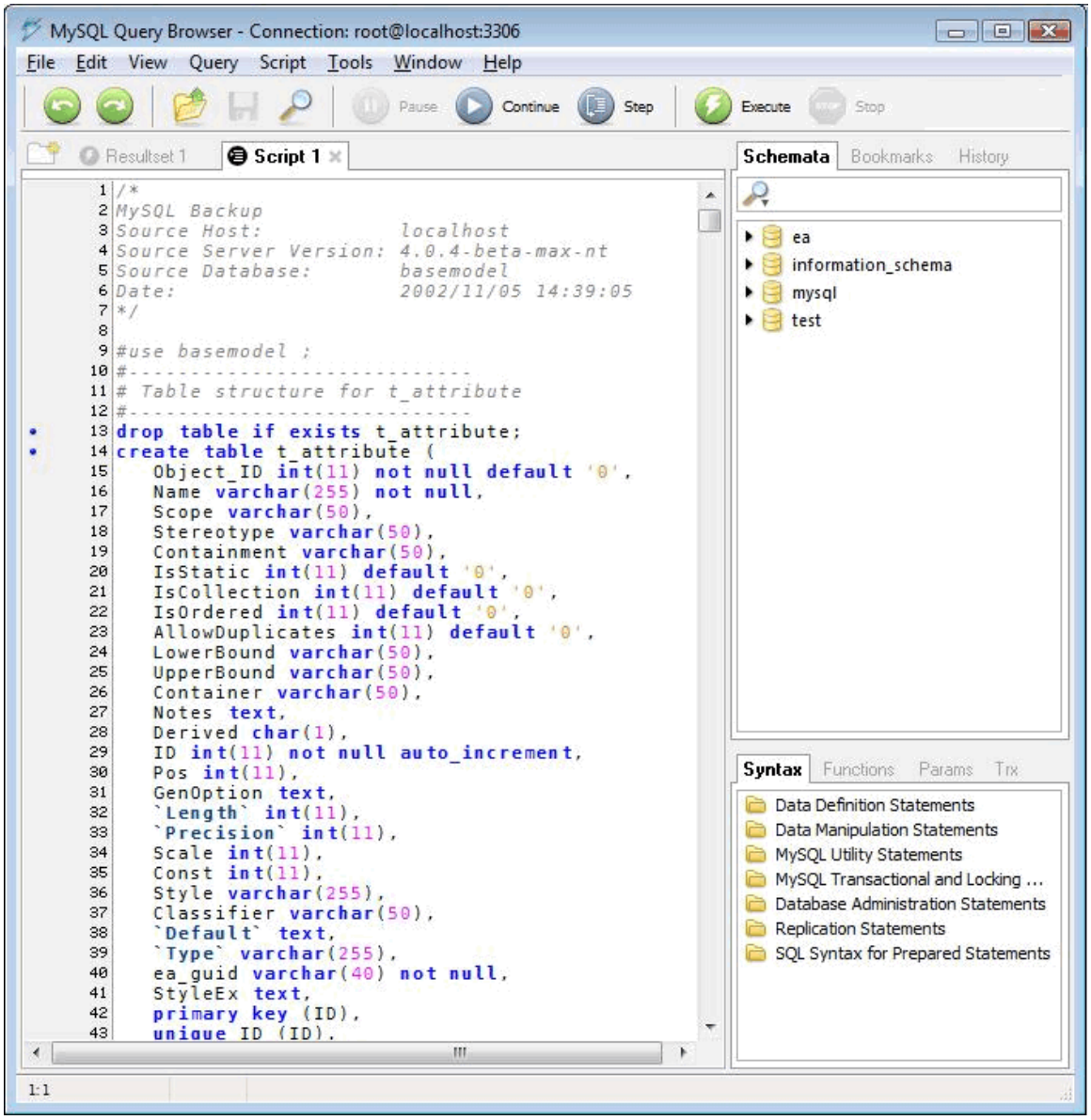
MySQL Administrator, available from http://dev.mysql.com/downloads/gui-tools/5.0.html, is one such tool. It provides a convenient graphical user interface to enable the creation of databases, execution of scripts, backups and restores.

To get started, follow the steps below:

1. Run MySQL Administrator and create a new database.



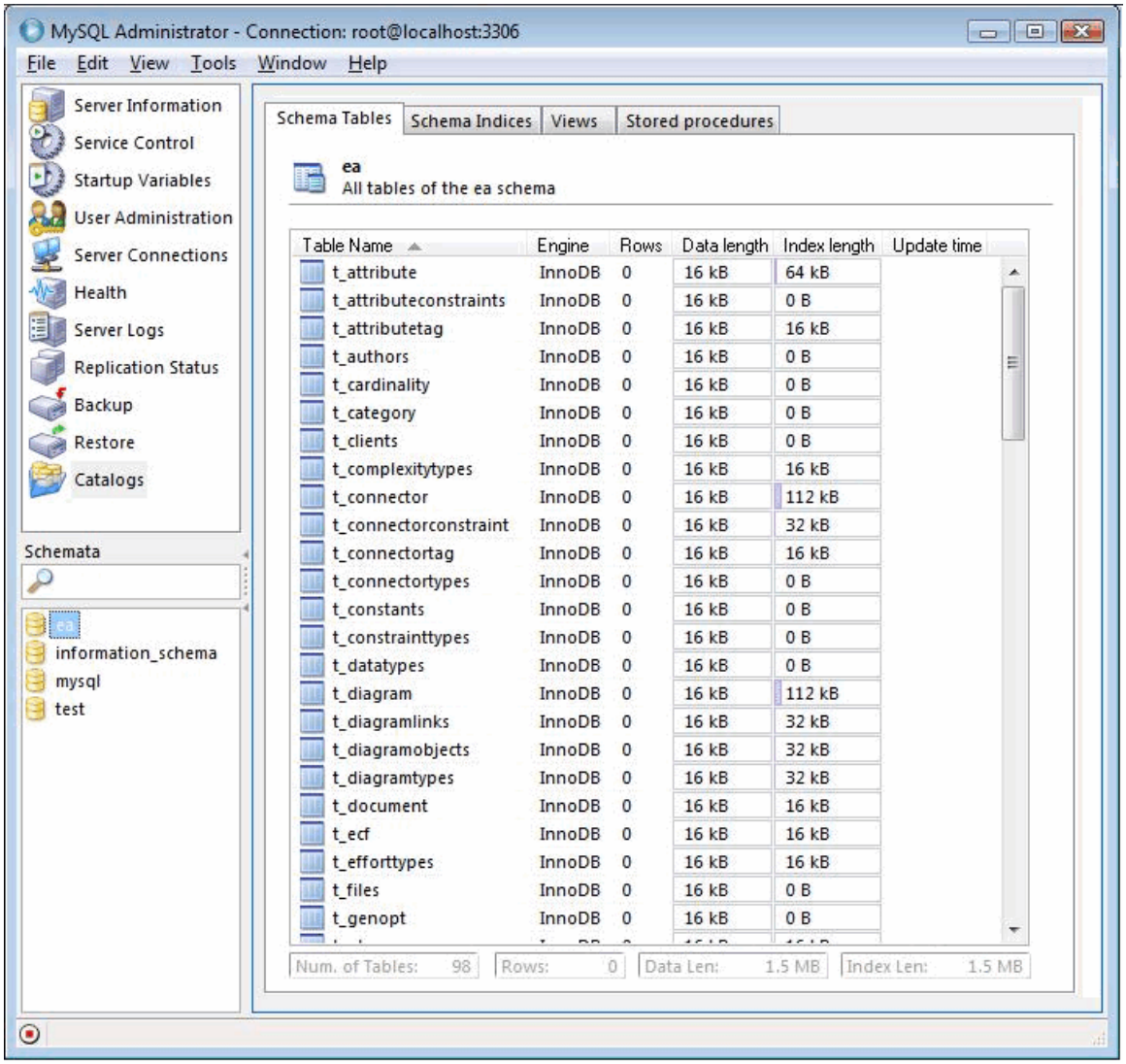
2. Run MySQL Query Browser, and open and execute the MySQL repository script.




```
/*
MySQL Backup
Source Host:           localhost
Source Server Version: 4.0.4-beta-max-nt
Source Database:       basemodel
Date:                  2002/11/05 14:39:05
*/

#use basemodel ;
#------------------------------
# Table structure for t_attribute
#------------------------------
drop table if exists t_attribute;
create table t_attribute (
   Object_ID int(11) not null default '0',
   Name varchar(255) not null,
   Scope varchar(50),
   Stereotype varchar(50),
   Containment varchar(50),
   IsStatic int(11) default '0',
   IsCollection int(11) default '0',
   IsOrdered int(11) default '0',
   AllowDuplicates int(11) default '0',
   LowerBound varchar(50),
   UpperBound varchar(50),
   Container varchar(50),
   Notes text,
   Derived char(1),
   ID int(11) not null auto_increment,
   Pos int(11),
   GenOption text,
   `Length` int(11),
   `Precision` int(11),
   Scale int(11),
   Const int(11),
   Style varchar(255),
   Classifier varchar(50),
   `Default` text,
   `Type` varchar(255),
   ea_guid varchar(40) not null,
   StyleEx text,
   primary key (ID),
   unique ID (ID),
```

3. Below is an example showing the tables created in the MySQL repository after running the script in MySQL Query Browser.



1.4.10.3 SQL Server Repository

Note:

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

Before creating a SQL Server data repository, you must have SQL Server and MDAC 2.6 or higher installed, and access permission to create a new database. Please note that setting up SQL Server and the issues involved are beyond the scope of this user guide. Consult your program's documentation for a guide to this.

Sparx Systems provide SQL scripts to create the required tables; how you create the database and execute that script are up to you.

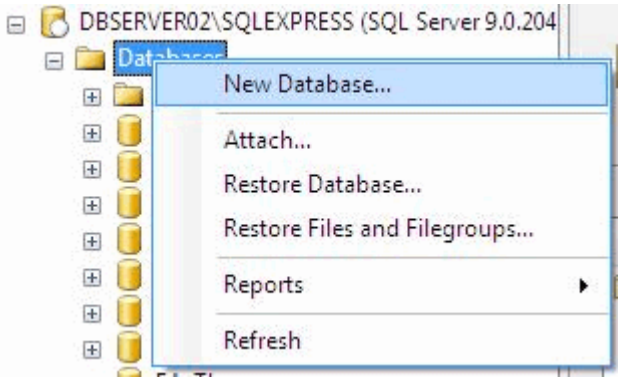
- Registered users can obtain the scripts from the Registered Corporate edition Resources page of the Sparx Systems website at http://www.sparxsystems.com/registered/reg_ea_corp_ed.html
- Trial users can obtain the scripts from the Corporate edition Resources page of the Sparx Systems website at http://www.sparxsystems.com/resources/corporate/.

Create a SQL Server Repository

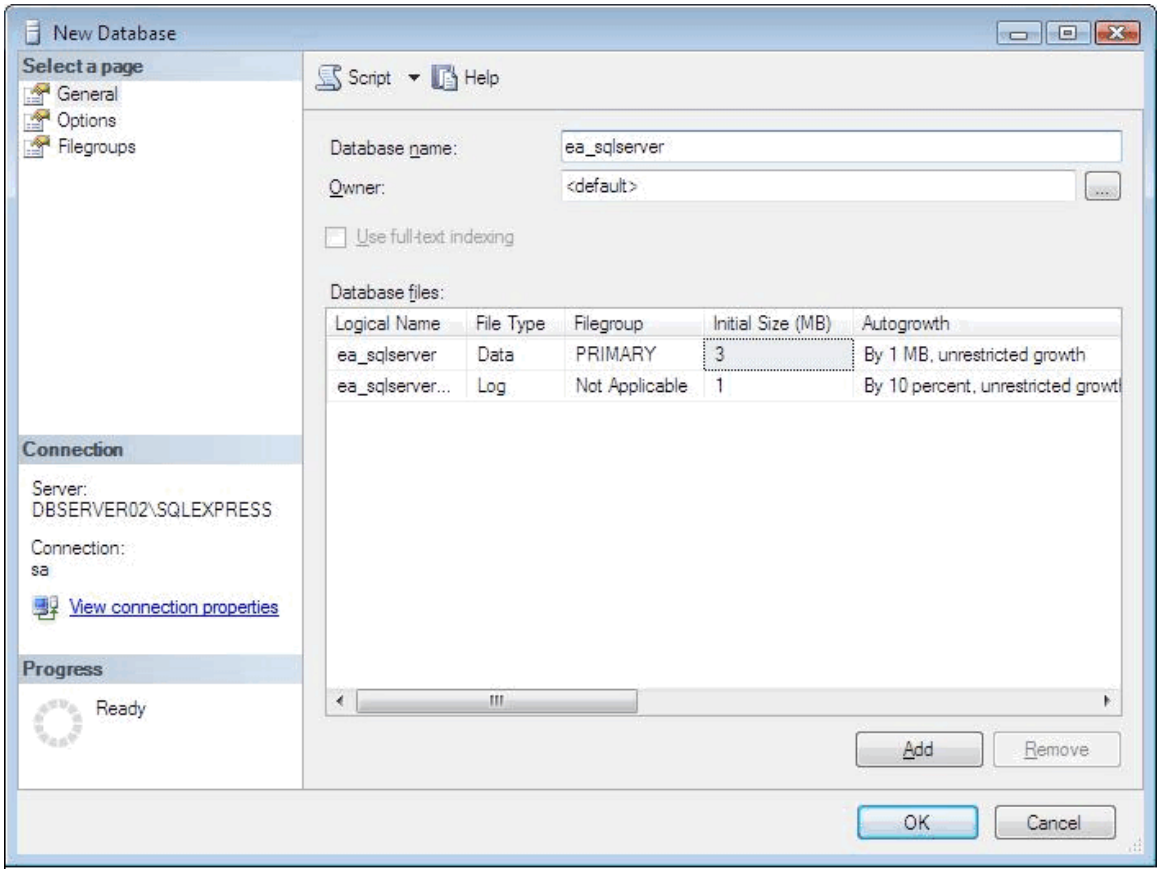
SQL Server repositories are created without any data, so you must perform a project data transfer 79 in Enterprise Architect to copy a suitable starter project. If you are starting from scratch, EABase.EAP 6 is a good starting point. If you are using an existing .EAP model, you can upsize 11 it.

To use SQL Enterprise Manager to create a SQL Server repository, follow the steps below:

1. In SQL Enterprise Manager, locate the server on which to create your new Enterprise Architect model; in the example below this is DBSERVER02\SQLEXPRESS.



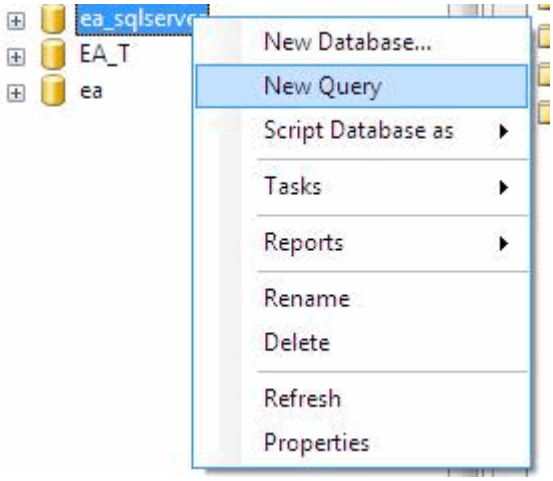
2. Right-click and choose the **New Database** context menu option.
3. Enter a suitable name for the database. Set any file options as required.



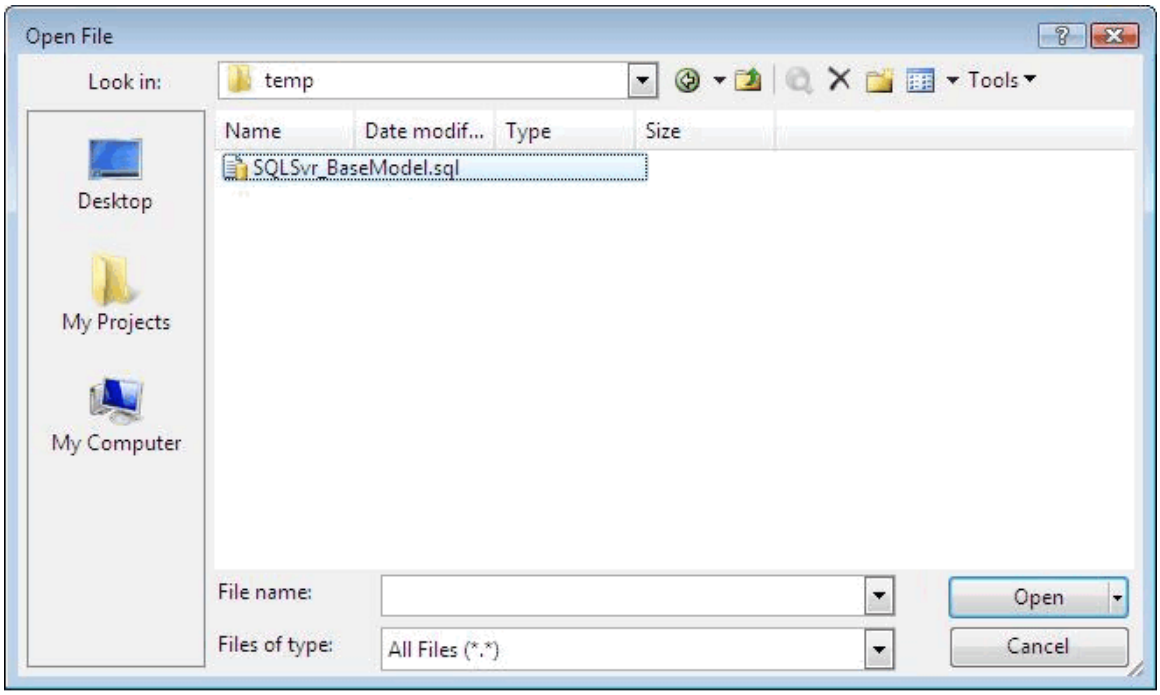
Note:

Ensure that the database collation is case-insensitive.

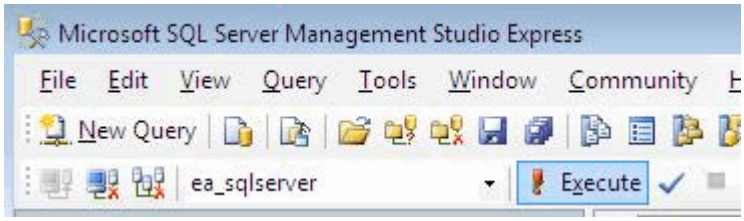
4. Click on the database to select it, then select the **New Query** menu option.



5. In the Query window, use the Open File dialog to locate the supplied Enterprise Architect SQL Server Model script file.



6. Click on the **Open** button. Check that you have selected the correct database to run the script in. In this example the tables are being added to the *ea_sqlserver* database as shown in the drop-down menu below.



7. Click on the **Execute** button; SQL Server executes the script, which creates the base model for an Enterprise Architect project.

1.4.10.4 Oracle Server Repository

Note:

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

Before creating an Oracle data repository, you must have the appropriate version of Oracle (9i, 10g or 11g) and MDAC 2.6 or higher installed, and access permission to create a new database. Please note that setting up Oracle and the issues involved are beyond the scope of this manual. Consult your program documentation for guidance.

Sparx Systems provide SQL scripts to create the required tables; how you create the database and execute that script are up to you.

- Registered users can obtain the scripts from the Registered Corporate edition Resources page of the Sparx Systems website at http://www.sparxsystems.com/registered/reg_ea_oracle_instructions.html
- Trial users can obtain the scripts from the Corporate edition Resources page of the Sparx Systems website at http://www.sparxsystems.com/resources/corporate/.

Create the Data Repository

Oracle repositories are created without any data, so you must perform a project data transfer 79 in Enterprise Architect to copy a suitable starter project. If you are starting from scratch, EABase.EAP 9 is a good starting point. If you want to use an existing .EAP model, you can upsize 19 it; follow the steps below:

1. Create a new database on the Oracle server.
2. Connect to the newly created database with a program such as Oracle SQL*Plus or SQL Plus Worksheet.
3. Execute the script Oracle_BaseModel.sql, which creates the base model tables and indexes for an Enterprise Architect Project.

Note:

Use the Project Data Transfer 79 function to upload a basic model into the repository.

1.4.10.5 PostgreSQL Repository

Note:

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

Before creating a PostgreSQL data repository in Enterprise Architect, you must set up PostgreSQL and PostgreSQL ODBC drivers. For further information on setting these up, see Set up a PostgreSQL ODBC Driver 27.

To create a new PostgreSQL repository, you must first create a database into which to import the table definitions for Enterprise Architect. Sparx Systems provide SQL scripts to create the required tables; how you create the database and execute that script are up to you.

- Registered users can obtain the scripts from the Registered Corporate edition Resources page of the Sparx Systems website at http://www.sparxsystems.com/registered/reg_ea_corp_ed.html
- Trial users can obtain the scripts from the Corporate edition Resources page of the Sparx Systems website at http://www.sparxsystems.com/resources/corporate/.

Create the Data Repository

After you create the database and execute the script, the result should be an empty Enterprise Architect project to begin working with. You can transfer data from an existing .EAP file or simply start from scratch.

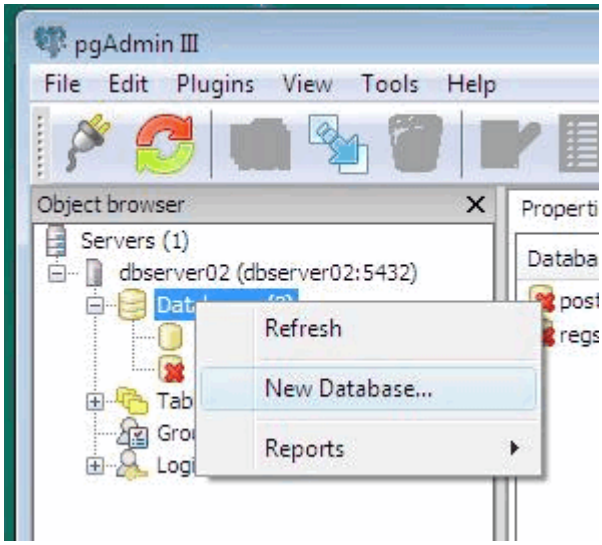
Third Party Tools

If you are unfamiliar with PostgreSQL and DBMS systems in general, you might want to consider a suitable front end tool.

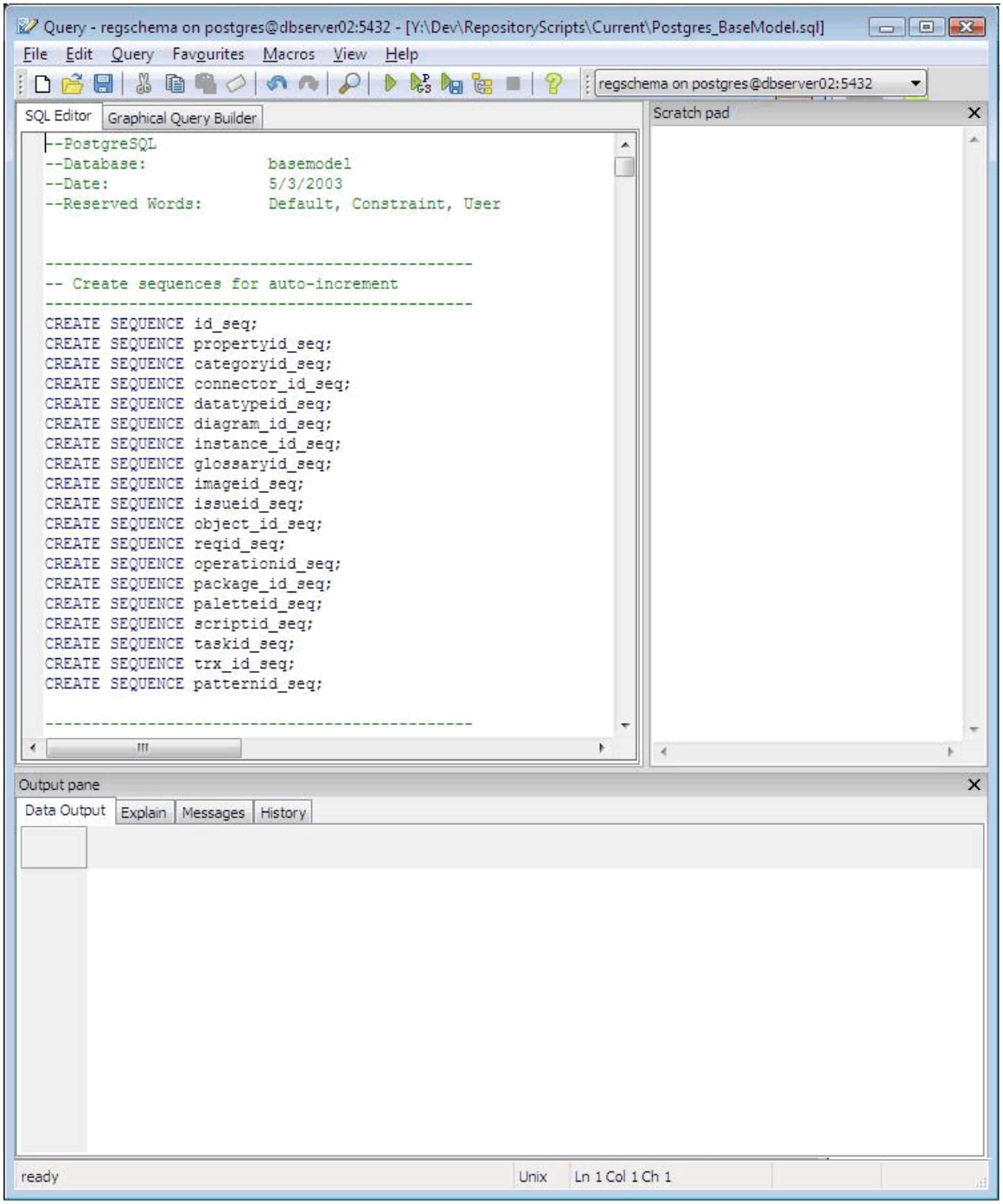
One such tool is *pgAdminIII*, which is available at http://www.pgadmin.org/download/. It provides a convenient graphical user interface to enable creation of databases, execution of scripts and restores.

To get started with pgAdminIII, follow the steps below:

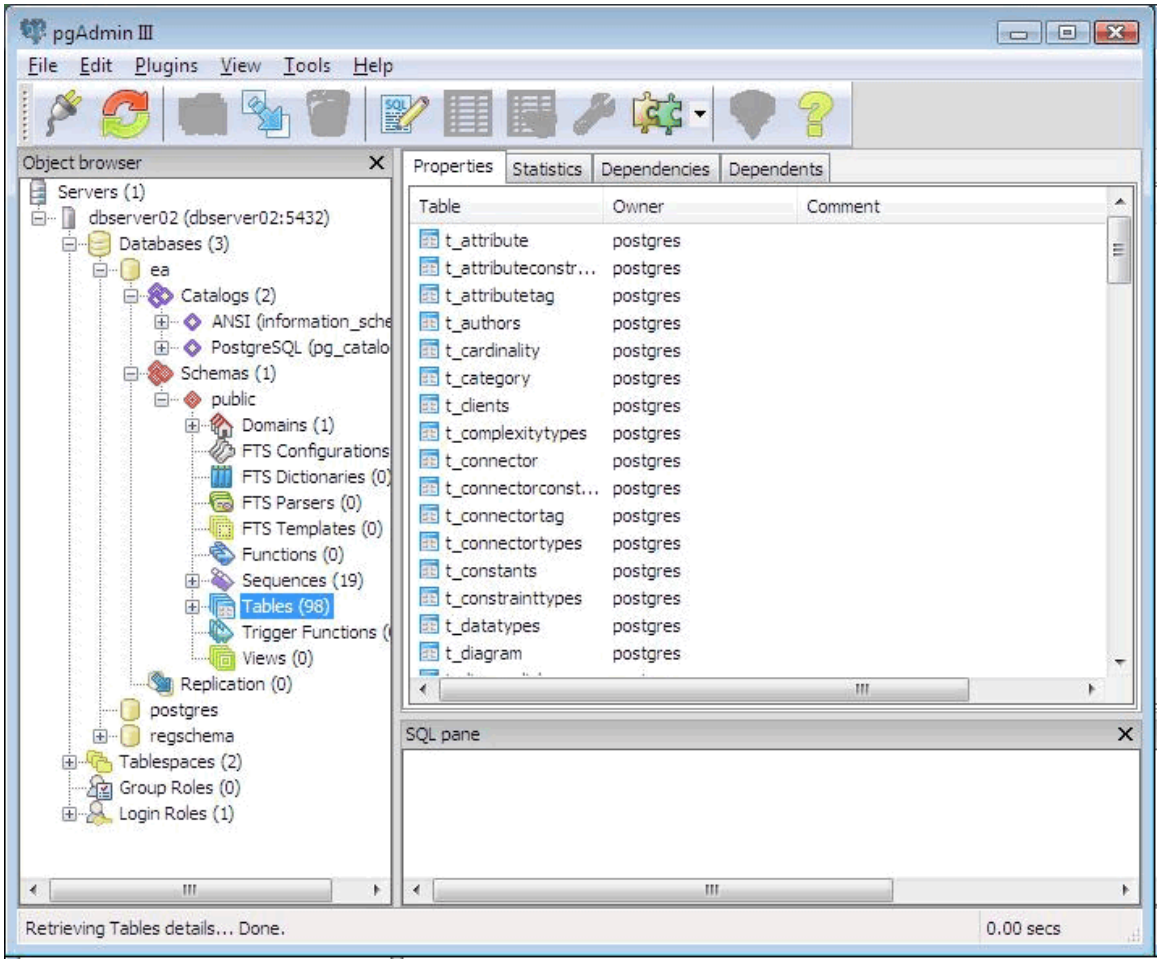
1. Create a new database.



2. Open and execute the PostgreSQL SQL script.



3. Below is an example showing the tables created in the PostgreSQL repository after running the script in pgAdminIII.



1.4.10.6 Adaptive Server Anywhere Repository

Note:

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

Before creating an ASA data repository in Enterprise Architect, you must set up ASA and ASA ODBC drivers. For further information on setting these up, see Setup an Adaptive Server Anywhere ODBC Driver 30.

To create a new ASA repository, you must first create a database into which to import the table definitions for Enterprise Architect. Sparx Systems provide SQL scripts to create the required tables - how you create the database and execute that script are up to you.

- Registered users can obtain the scripts from the Registered Corporate edition Resources page of the Sparx Systems website at http://www.sparxsystems.com/registered/reg_ea_corp_ed.html
- Trial users can obtain the scripts from the Corporate edition Resources page of the Sparx Systems website at http://www.sparxsystems.com/resources/corporate/.

Create the Data Repository

After you create the database and execute the script, the result should be an empty Enterprise Architect project to begin working with. You can transfer data from an existing .EAP file or simply start from scratch.

Third Party Tools

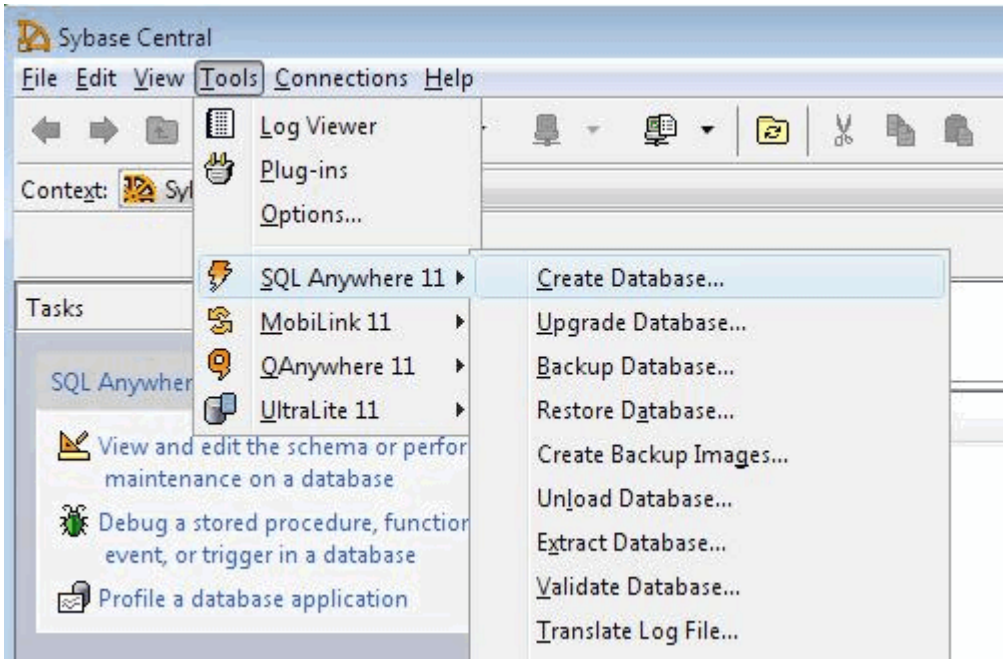
If you are unfamiliar with ASA and DBMS systems in general, you might want to consider a suitable front end tool.

Sybase Central is one such tool, that can be installed along with the DBMS. It provides a convenient graphical

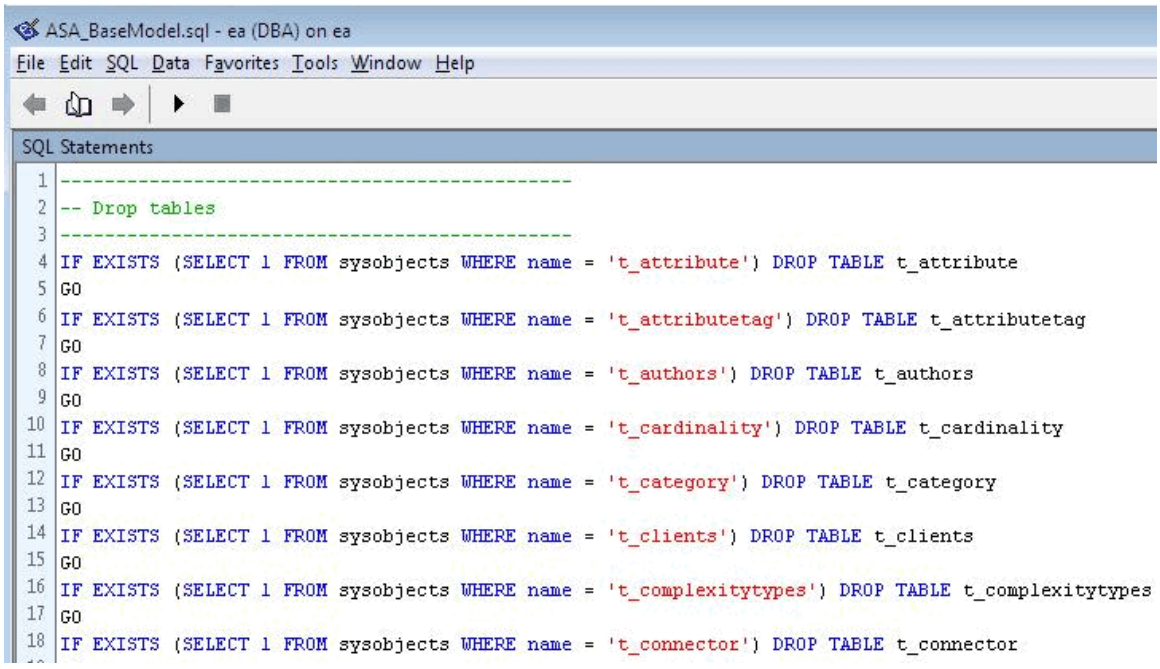
user interface to enable creation of databases, execution of scripts and restores.

To get started with Sybase Central, follow the steps below:

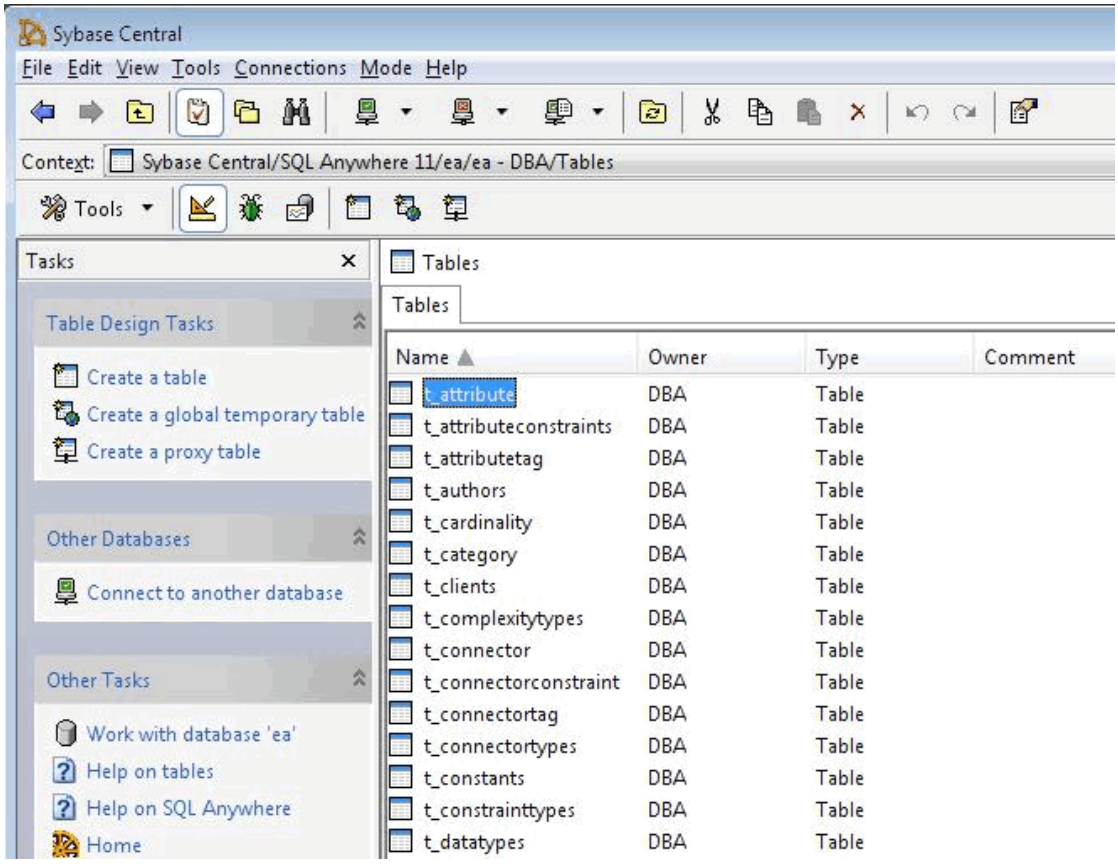
1. Create a new database.



2. Open Interactive SQL and execute the AS SQL script.



The following example shows the tables created in the ASA repository after running the script in EMS ASA Manager.



1.4.10.7 MSDE Server Repository

Note:

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

Before creating a SQL Server MSDE data repository, you must have MSDE Server and MDAC 2.6 or higher installed. Please note that setting up MSDE Server and the issues involved are beyond the scope of this user guide. Consult your program documentation for guidance.

Sparx Systems provide SQL scripts to create the required tables; how you create the database and execute that script are up to you.

- Registered users can obtain the scripts from the Registered Corporate edition Resources page of the Sparx Systems website at http://www.sparxsystems.com/registered/reg_ea_corp_ed.html
- Trial users can obtain the scripts from the Corporate edition Resources page of the Sparx Systems website at http://www.sparxsystems.com/resources/corporate/.

Use the SQL Server 2000, 2005 or 2008 script for MSDE, and follow the steps to Create a New SQL Server Data Repository 40.

1.4.10.8 Progress OpenEdge Repository

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions.
- The OpenEdge database must be either version 10.0B03 or version 10.1B01, or later.

Before creating a Progress OpenEdge data repository, you must have OpenEdge and MDAC 2.6 or higher

installed, and access permission to create a new database. Please note that setting up OpenEdge and the issues involved are beyond the scope of this manual. Consult your OpenEdge documentation for guidance.

Sparx Systems provide SQL scripts to create the required tables; how you create the database and execute that script is up to you.

- Registered users can obtain the scripts from the Registered Corporate edition Resources page of the Sparx Systems website at http://www.sparxsystems.com/registered/reg_ea_openedge_instructions.html
- Trial users can obtain the scripts from the Corporate edition Resources page of the Sparx Systems website at http://www.sparxsystems.com/resources/corporate/.

Create the Data Repository

OpenEdge repositories are created without any data, so you must perform a project data transfer 79 in Enterprise Architect to copy a suitable starter project. If you are starting from scratch, EABase.EAP 9 is a good starting point. If you want to use an existing .EAP model, you can upsize 15 it.

1. Run proenv from the **OpenEdge** menu: **Start->Programs->OpenEdge->proenv**.
2. Create database: *prodb <database_name>* empty.
3. Start database server: **proserve <database_name> -S <port_number>**
4. Open Data Administration to add a user:
```
prowin32 -db <database_name> -S <port_number> -p _admin -rx
```
5. Open **Admin->Security->Edit User List**.
6. Close Data Administration.
7. Open SQL Explorer Tool, connect as 'sysprogress'.
8. Add user:
```
create user 'user','password';
commit;
```
9. Grant privileges:
```
grant dba, resource to <user>;
commit;
```

Tip:

Use the Project Data Transfer 79 function to upload a basic model into the repository.

1.4.11 Connect to a Data Repository

In Enterprise Architect you connect to a data repository for one of two reasons:

- To access an existing Enterprise Architect model in the repository
- To reverse engineer a database schema into a model using ODBC (see *Code Engineering Using UML Models*).

Note:

To connect to a repository, you must have the usual SELECT, UPDATE, INSERT and DELETE permissions.

This topic describes how to connect to the following data repositories, to access an existing Enterprise Architect model:

- MySQL Data Repository 50
- SQL Server Data Repository 52
- Oracle Data Repository 55
- PostgreSQL Data Repository 62
- Adaptive Server Anywhere Data Repository 65
- MSDE Server Data Repository 68
- Progress OpenEdge 68.

1.4.11.1 MySQL Data Repository

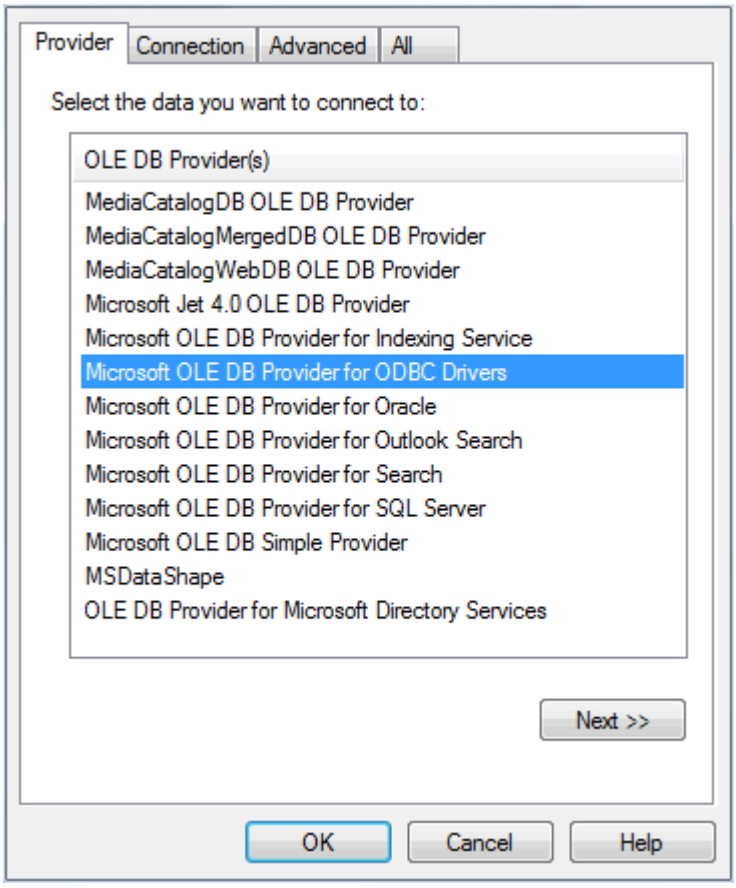
Note:

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

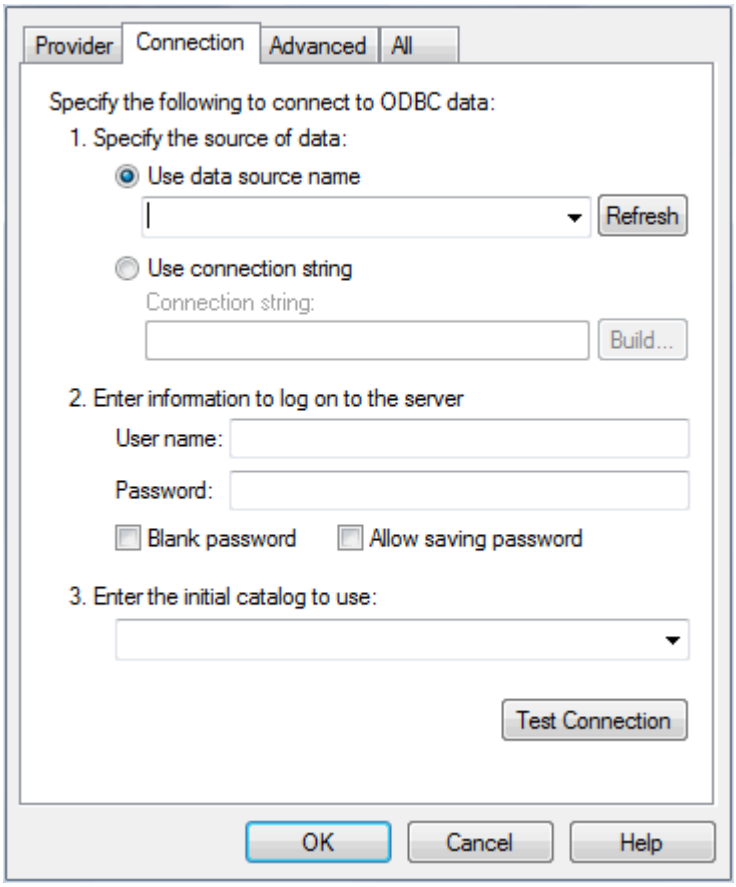
In order to use a MySQL data repository, you must connect to it in Enterprise Architect first. Before connecting to the repository, you must set up a MySQL ODBC driver 25.

To connect to a MySQL data repository in Enterprise Architect, follow the steps below:

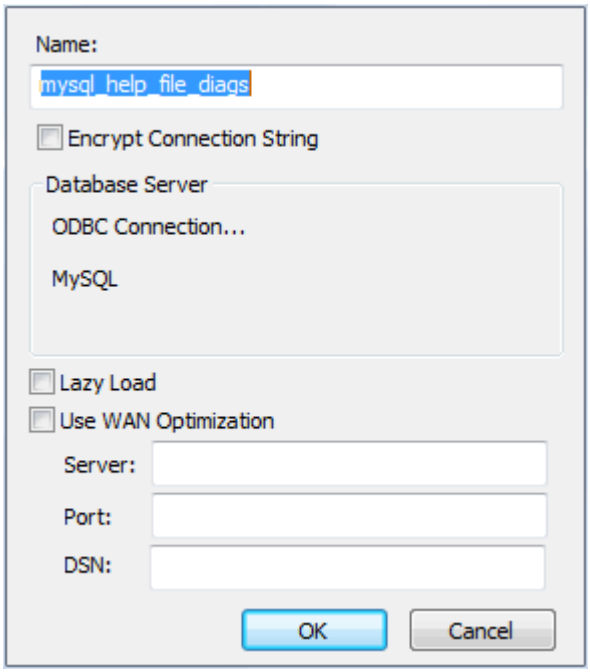
1. In the Open Project 8 dialog, select the **Connect to Server** checkbox.
2. Click on the **[...]** (Browse) button, as you normally would to browse for a project. As you have selected the **Connect to Server** checkbox, the Data Link Properties dialog displays instead of the Browse Directories dialog.



3. Select **Microsoft OLE DB Provider for ODBC Drivers** from the list.
4. Click on the **Next** button. The Connection tab displays.



5. Click on the **Use data source name** radio button and on the drop-down arrow in its field. Select the ODBC driver you have set up to connect to your MySQL repository from the list. In the setup example 25 the driver title is **MySQL-EABASE**.
6. If required, type in a **User name** and **Password**.
7. If required, type in an initial catalog.
8. Click on the **Test Connection** button to confirm that the details are correct.
9. If the test succeeds, click on the **OK** button.
10. If the test does not succeed, revise your settings.
11. After you have clicked on the **OK** button, the Connection Name & Type dialog displays.



12. Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog 8.
13. If required, select the **Encrypt Connection String** checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
14. If required, select the **Lazy Load** checkbox to not load the full project view when the model is loaded. Instead, only the parts that are necessary to display the visible portion of the tree are loaded. This means that a model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.
15. If required, select the **Use WAN Optimization** 72 checkbox. (To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time.)

    If you select this checkbox, complete the next three fields (see your administrator for the correct values). Otherwise go to step 19.
16. In the **Server** field, type the network name or address of the optimizer server.
17. In the **Port** field, type the port on which the server is running on the remote machine.
18. In the **DSN** field, type the data source name of the database as it appears on the remote machine.
19. Click on the **OK** button to complete the configuration.

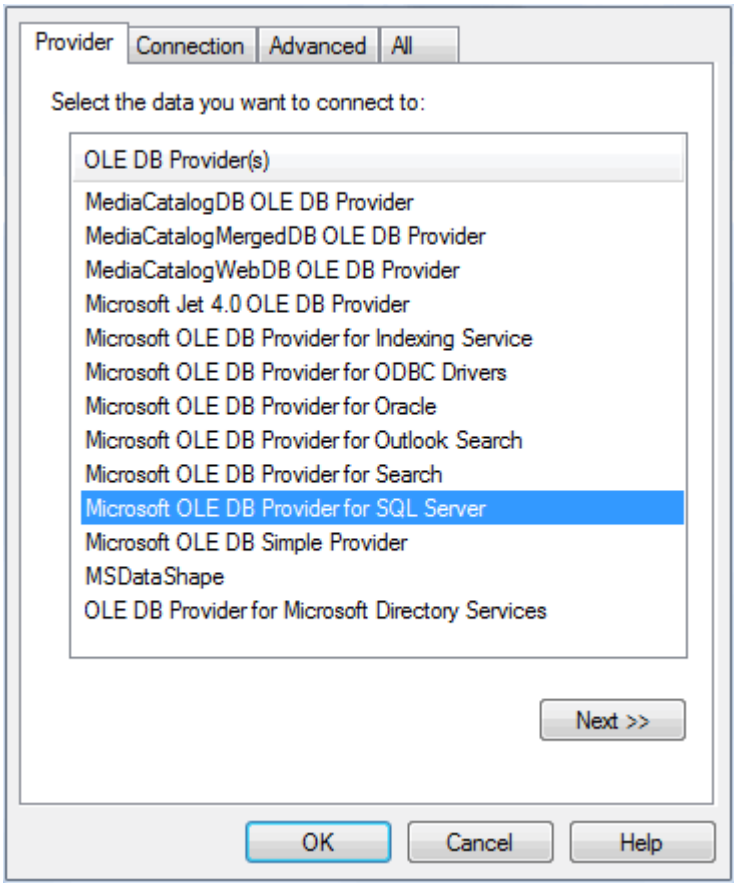
1.4.11.2 SQL Server Data Repository

Note:

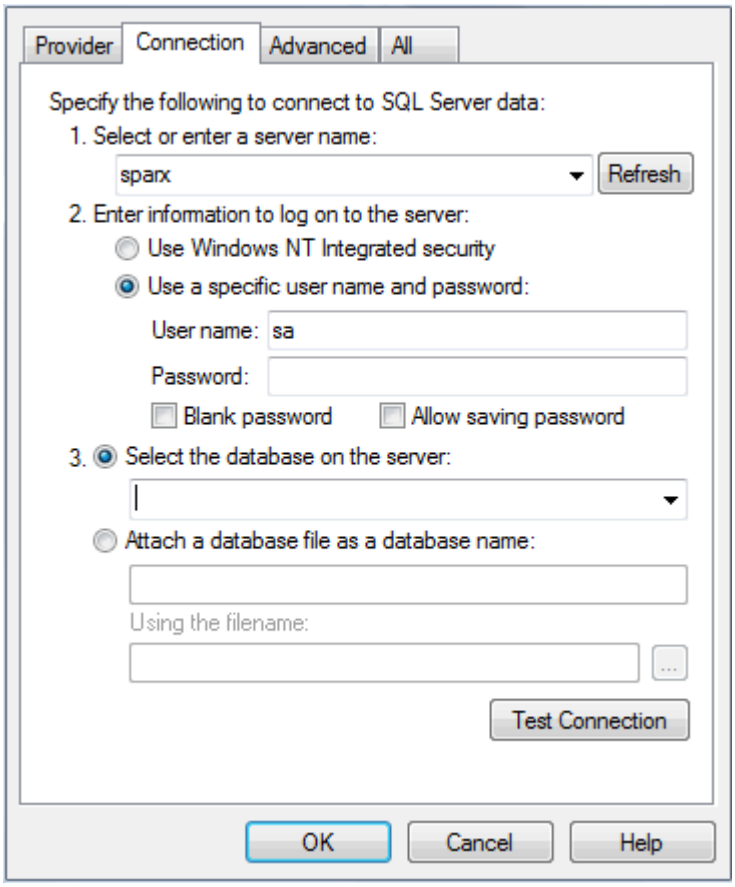
This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions.

Before you can use a SQL Server data repository, you must connect to it in Enterprise Architect. To connect to your SQL Server data repository in Enterprise Architect, follow the steps below:

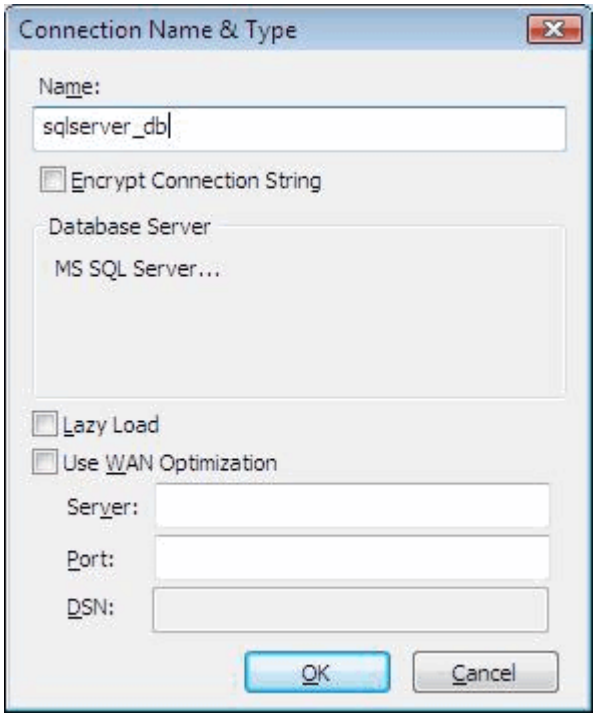
1. In the Open Project dialog 8, select the **Connect to Server** checkbox.
2. Click on the **[...]** (Browse) button, as you normally would to browse for a project. As you have selected the **Connect to Server** checkbox, the Data Link Properties dialog displays instead of the Select Enterprise Architect Project to Open dialog.



3. Select **Microsoft OLE DB Provider for SQL Server** from the list.
4. Click on the **Next>>** button. The Connection tab displays.



5. Type in the server details, including **Server Name**, **User Name** and **Password**.
6. Click on the **Select the database on the server** option and on the drop-down arrow. From the list, select the model to connect to.
7. Click on the **Test Connection** button to confirm that the details are correct.
8. If the test succeeds, click on the **OK** button. If the test does not succeed, revise your settings.
9. When you click on the **OK** button, the Connection Name & Type dialog displays.



10. In the **Name** field, type a suitable name for the connection so that you can recognize it in the Recent Projects panel on the Open Project 8 dialog.
11. If required, select the **Encrypt Connection String** checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
12. If required, select the **Lazy Load** checkbox to not load the full project view when the model is loaded. Instead, only the parts that are necessary to display the visible portion of the tree are loaded. This means that a model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.
13. If required, select the **Use WAN Optimization** 72 checkbox. (To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time.)

    If you select this checkbox, complete the next two fields (see your administrator for the correct values). Otherwise go to step 16.
14. In the **Server** field, type the network name or address of the optimizer server.
15. In the **Port** field, type the port on which the server is running on the remote machine.
16. Click on the **OK** button to complete the configuration.

1.4.11.3 Oracle Data Repository

Note:

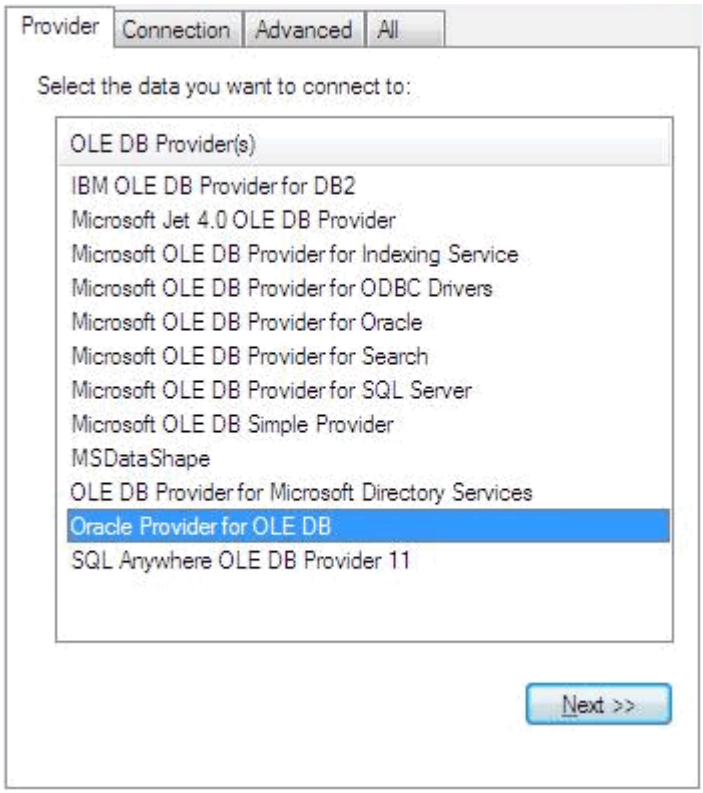
This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions.

In order to use an Oracle 9i, 10g or 11g data repository, you must connect to it in Enterprise Architect first. You can connect using either OLE DB 55 or ODBC 58.

Using Oracle OLE DB Provider

To connect to your Oracle data repository using the Oracle OLE DB Provider, follow the steps below:

1. In the Open Project dialog 8, select the **Connect to Server** checkbox.
2. Click on the **[...]** (Browse) button, as you normally would to browse for a project. As you have selected the **Connect to Server** checkbox, the Data Link Properties dialog displays instead of the Browse Directories dialog.

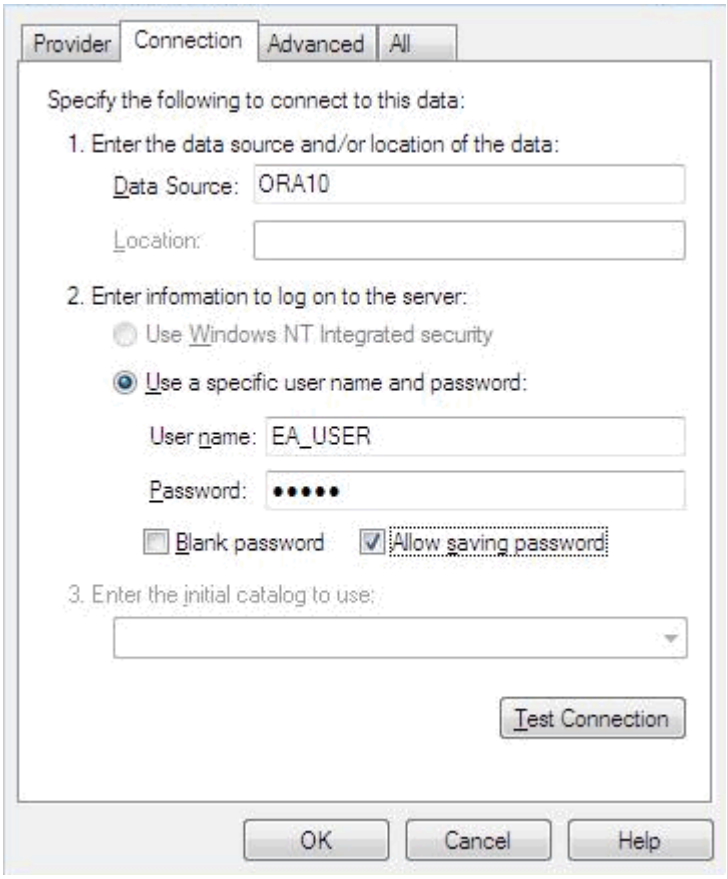


3. Select **Oracle Provider for OLE DB** from the list.

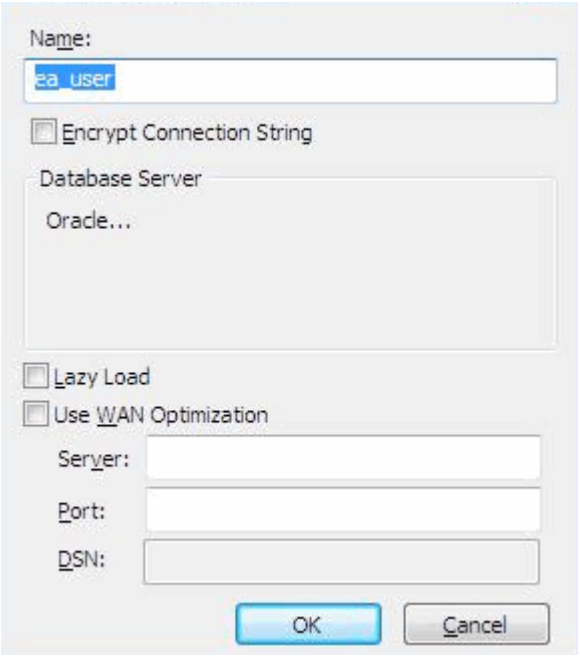
Note:

Do *not* select **Microsoft OLE DB Provider for Oracle**; Enterprise Architect might not work as expected.

4. Click on the **Next** button. The Connection tab displays.



5. Enter the **Data Source** name (the service name of the Oracle database), the database **User Name** and the **Password**. The **Location** field is not required.
6. Click on the **Test Connection** button to confirm that the details are correct.
7. If your test succeeded, click on the **OK** button.
8. If your test did not succeed, revise your settings.
9. After you have clicked on the **OK** button, the Connection Name and Type dialog displays.



10. Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog 8.
11. If required, select the **Encrypt Connection String** checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
12. If required, select the **Lazy Load** checkbox to not load the full project view when the model is loaded. Instead, only the parts that are necessary to display the visible portion of the tree are loaded. This means that a model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.
13. If required, select the **Use WAN Optimization** 72 checkbox. (To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time.)

    If you select this checkbox, complete the next two fields (see your administrator for the correct values). Otherwise go to step 16.
14. In the **Server** field, type the network name or address of the optimizer server.
15. In the **Port** field, type the port on which the server is running on the remote machine.
16. Click on the **OK** button to complete the configuration.

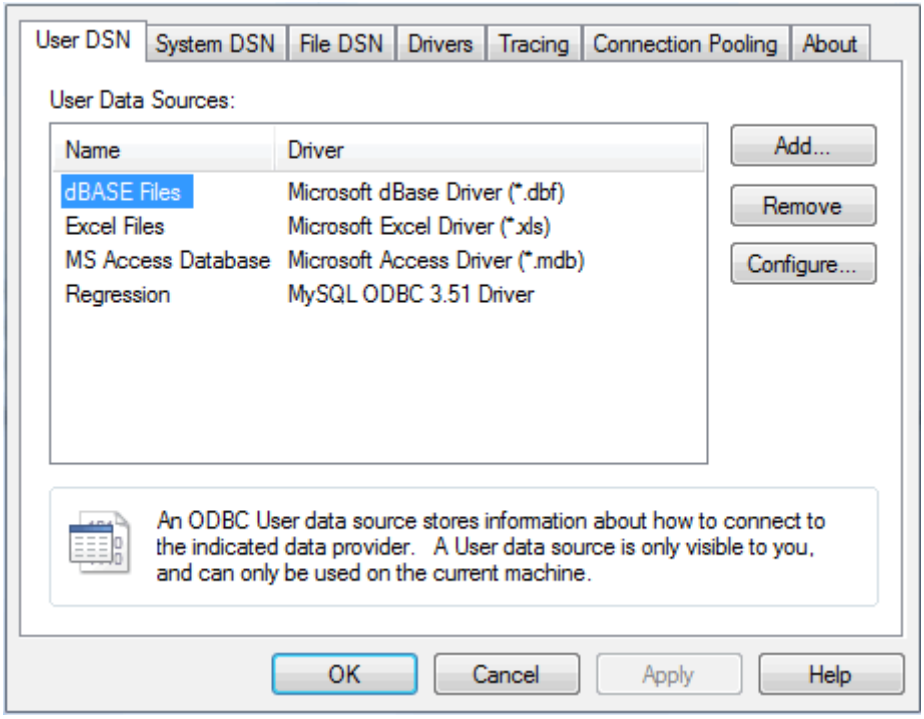
Using Oracle ODBC Driver

This process has two stages:

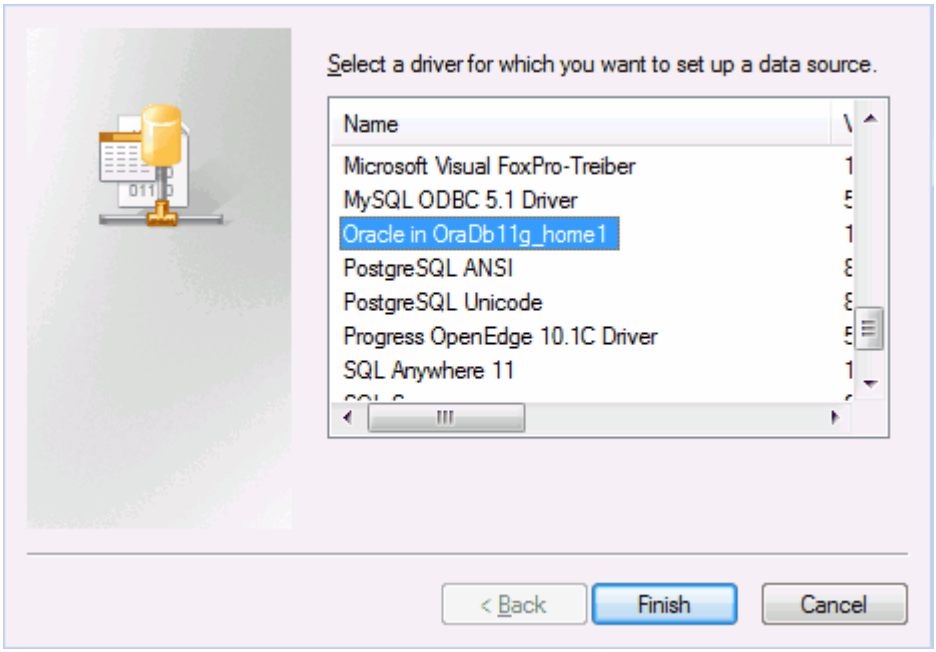
- Create ODBC Driver connection in Windows
- Connect to Repository in Enterprise Architect.

Create ODBC Driver Connection

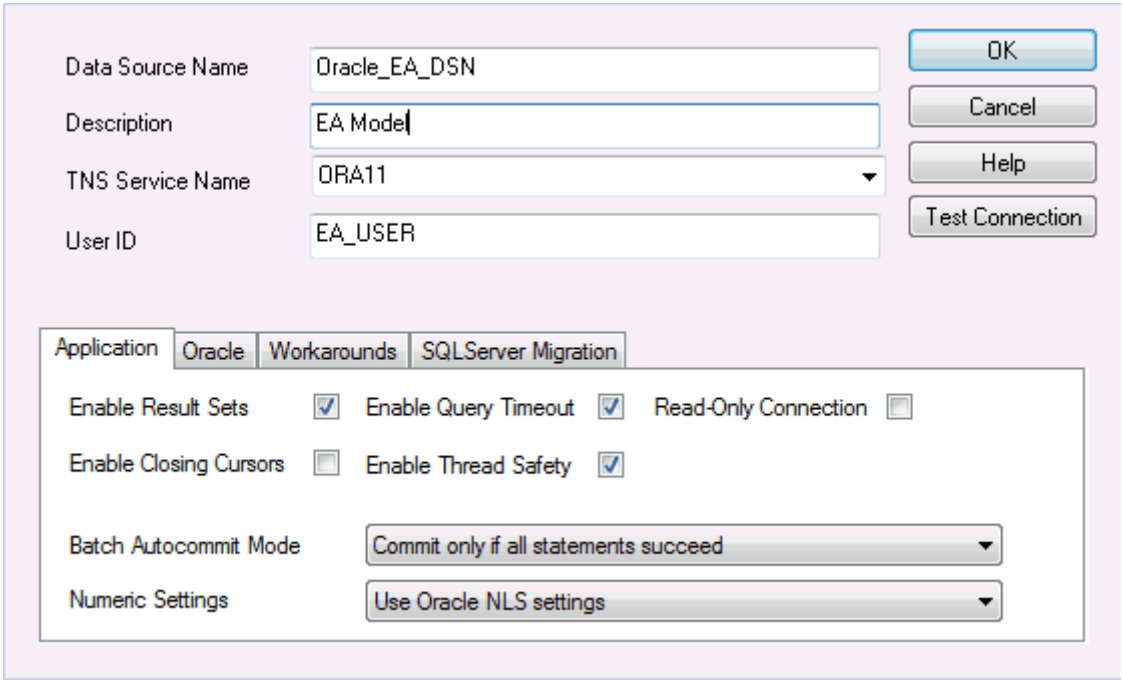
1. Select the Windows™ **Control Panel | Administrative Tools | Data Sources (ODBC)** option. The ODBC Data Source Administrator window displays.



2. Click on the **Add** button. The Create New Data Source dialog displays, enabling you to add a new DSN.



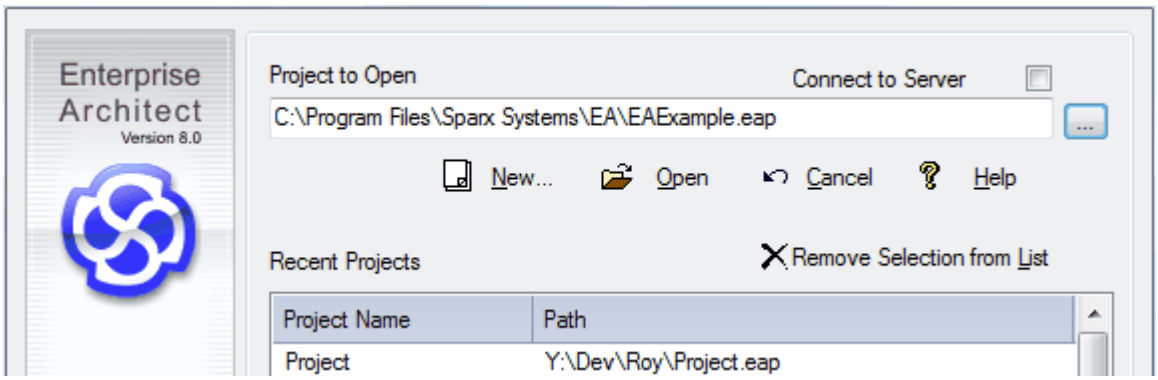
3. Select **Oracle in OraDB11g_home1** (or similar, depending on the ODBC installation).
4. Click on the **Finish** button. The Oracle ODBC Driver Configuration dialog displays.



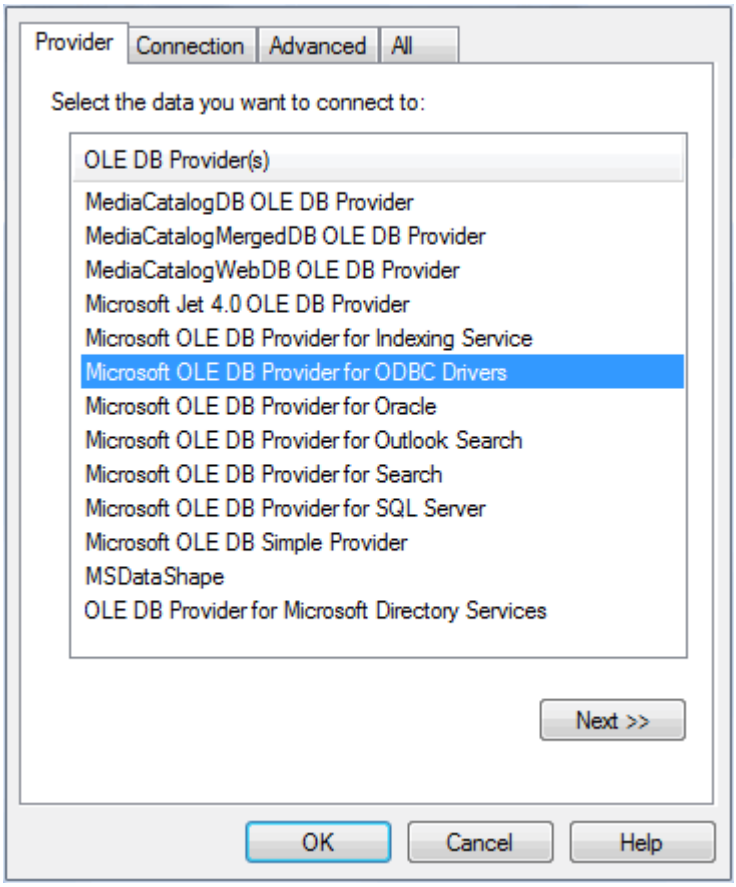
5. Enter the following configuration details:
   - A data source name for the connection
   - A description (optional)
   - The TNS Service Name (click on the drop-down arrow and select from the list)
   - The User ID.
6. Click on the **Test Connection** button and enter the Oracle user password to confirm that the details are correct.
7. Click on the **OK** button to complete the ODBC connection.

Connect To Repository

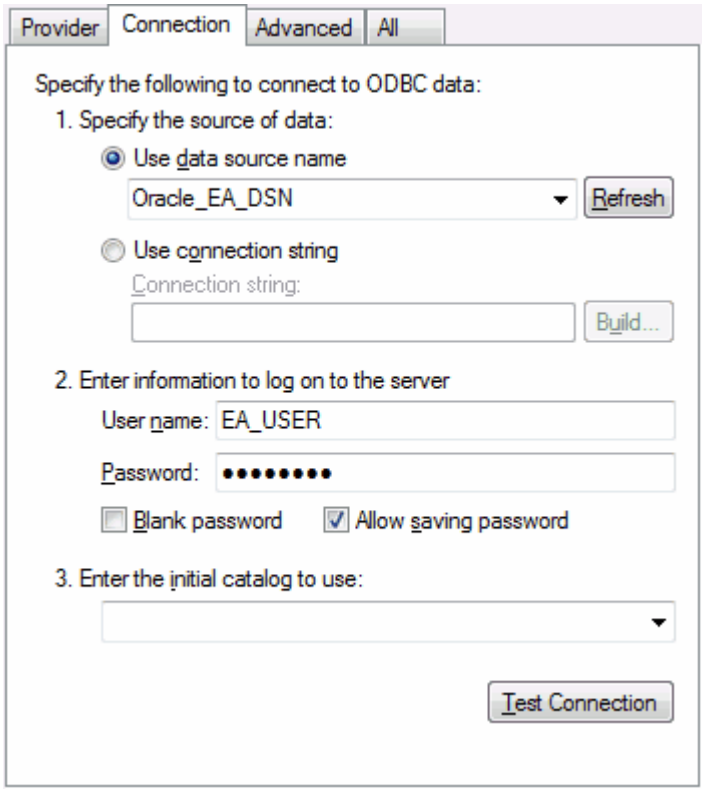
1. In Enterprise Architect, select the **File | Open Project** menu option. The Open Project dialog displays.



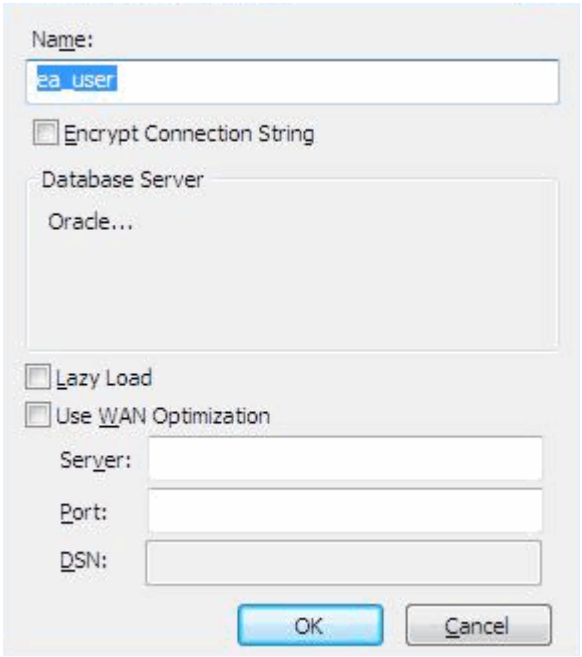
2. Select the **Connect to Server** checkbox.
3. Click on the **[...]** (Browse) button, as you normally would to browse for a project. As you have selected the **Connect to Server** checkbox, the Data Link Properties dialog displays instead of the Browse Directories dialog.



4. Select **Microsoft OLE DB Provider for ODBC Drivers** from the list.
5. Click on the **Next** button. The Connection tab displays.



6. Select the **Data Source** name from the drop-down list, and type in the database **User Name** and **Password**.
7. Click on the **Test Connection** button to confirm that the details are correct.
8. If your test succeeded, click on the **OK** button; if your test did not succeed, revise your settings.
9. After you have clicked on the **OK** button, Oracle prompts you for the password. Enter this. The Connection Name and Type dialog displays.



10. Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog 8.

11. If required, select the **Encrypt Connection String** checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
12. If required, select the **Lazy Load** checkbox to not load the full project view when the model is loaded. Instead, only the parts that are necessary to display the visible portion of the tree are loaded. This means that a model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.
13. If required, select the **Use WAN Optimization** 72 checkbox. (To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time.)

    If you select this checkbox, complete the next two fields (see your administrator for the correct values). Otherwise go to step 16.
14. In the **Server** field, type the network name or address of the optimizer server.
15. In the **Port** field, type the port on which the server is running on the remote machine.
16. Click on the **OK** button to complete the configuration.

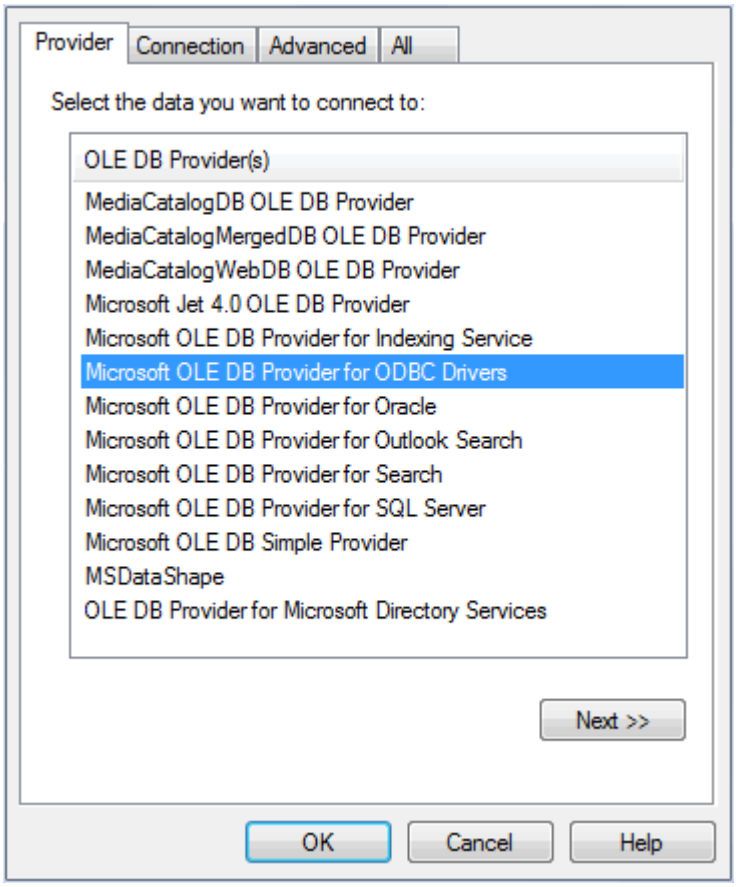
1.4.11.4 PostgreSQL Data Repository

Note:

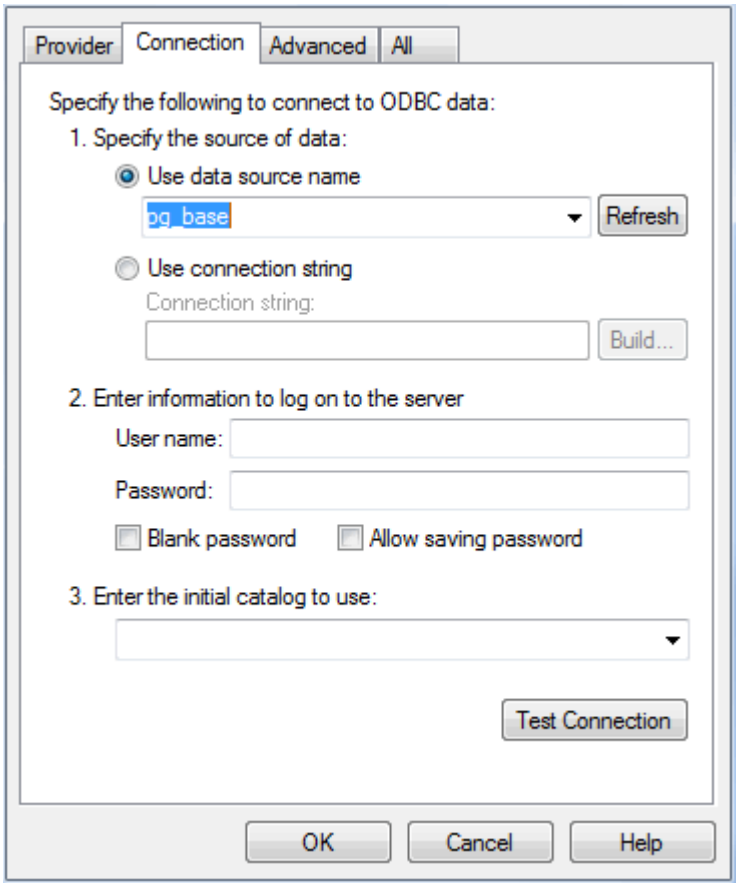
This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions.

In order to use a PostgreSQL data repository, you must connect to it in Enterprise Architect first. Before connecting to the repository, you must have set up a PostgreSQL ODBC driver 27. To connect to a PostgreSQL data repository in Enterprise Architect, follow the steps below:

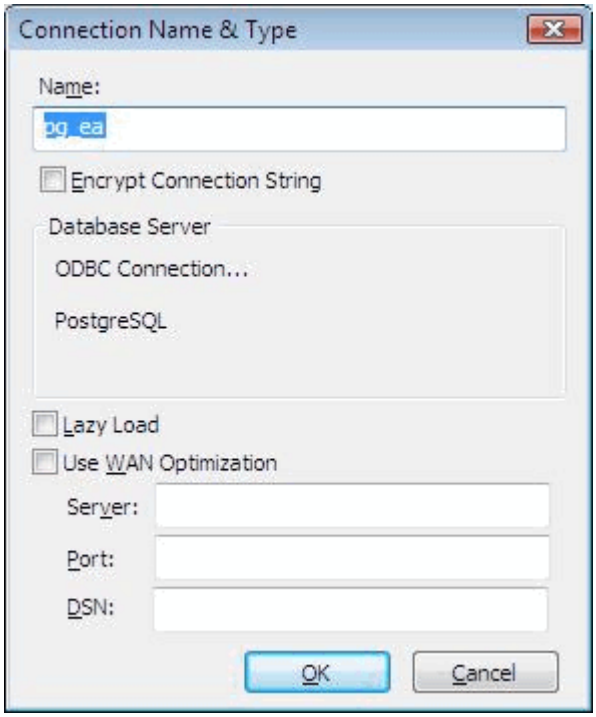
1. In the Open Project dialog 8, select the **Connect to Server** checkbox, or on the Start Page, click on the **Connect to Server** link.
2. Click on the **[...]** (Browse) button, as you normally would to browse for a project. As you have selected the **Connect to Server** checkbox, the Data Link Properties dialog displays instead of the Browse Directories dialog.



3. Select **Microsoft OLE DB Provider for ODBC Drivers** from the list.
4. Click on the **Next** button. The Connection tab displays.



5. Click on the **Use data source name** drop-down arrow and, from the list, select the ODBC driver you have set up to connect to your PostgreSQL repository.
6. Click on the **Test Connection** button to confirm that the details are correct.
7. If your test succeeded, click on the **OK** button.
8. If your test did not succeed, revise your settings.
9. After you have clicked on the **OK** button, the Connection Name & Type dialog displays.



10. Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog 8.
11. If required, select the **Encrypt Connection String** checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
12. If required, select the **Lazy Load** checkbox to not load the full project view when the model is loaded. Instead, only the parts that are necessary to display the visible portion of the tree are loaded. This means that a model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.
13. If required, select the **Use WAN Optimization** 72 checkbox. (To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time.)

    If you select this checkbox, complete the next three fields (see your administrator for the correct values). Otherwise go to step 17.
14. In the **Server** field, type the network name or address of the optimizer server.
15. In the **Port** field, type the port on which the server is running on the remote machine.
16. In the **DSN** field, type the data source name of the database as it appears on the remote machine.
17. Click on the **OK** button to complete the configuration.

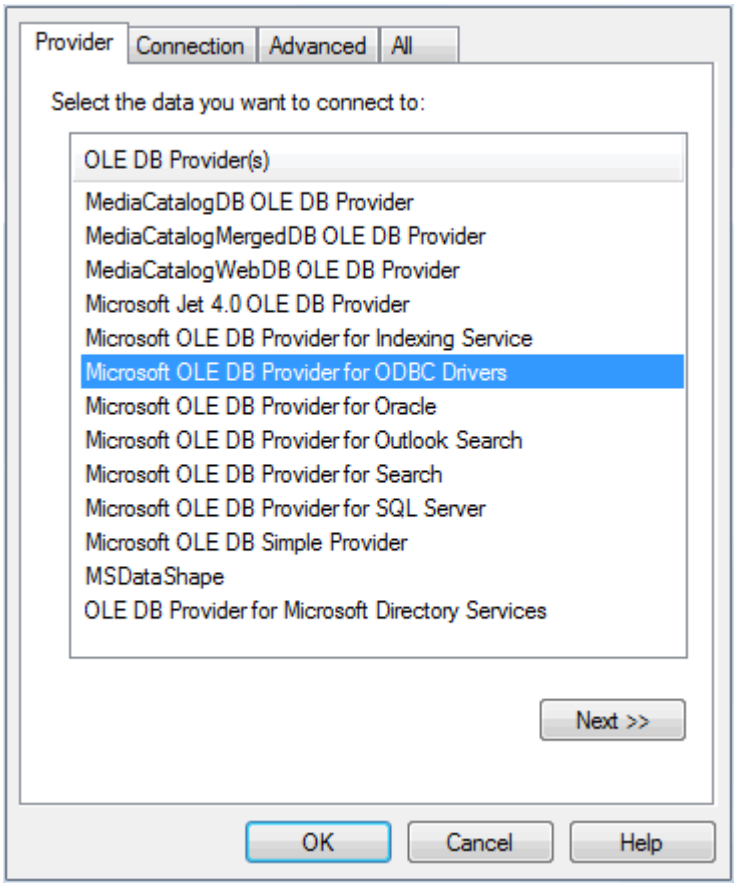
1.4.11.5 ASA Data Repository

Note:

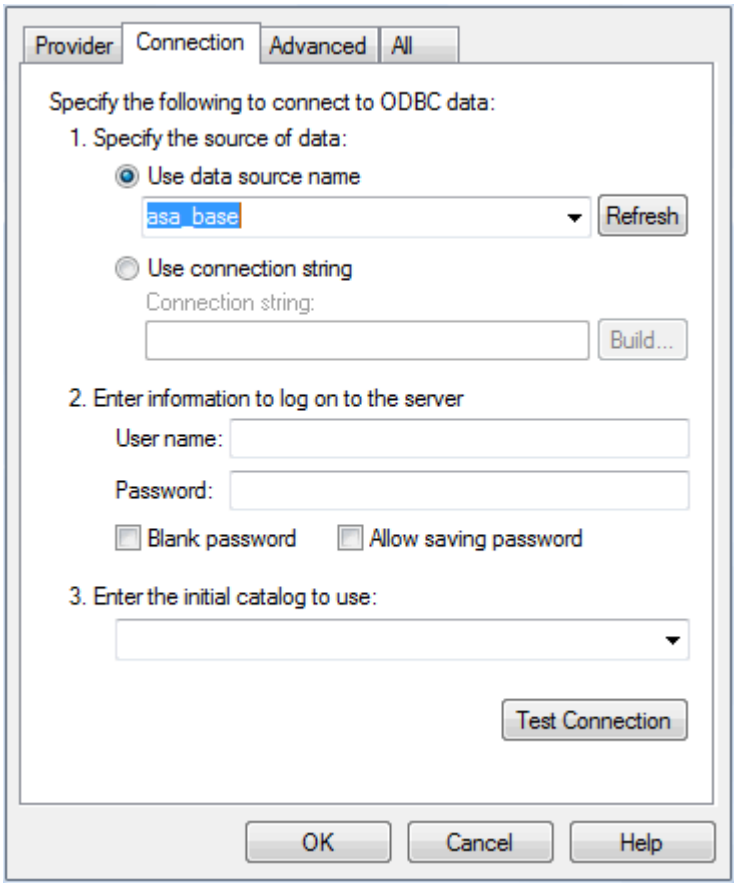
This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions.

In order to use an ASA data repository, you must connect to it in Enterprise Architect first. Before connecting to the repository, you must have set up an ASA ODBC driver 30. To connect to an ASA data repository in Enterprise Architect, follow the steps below:

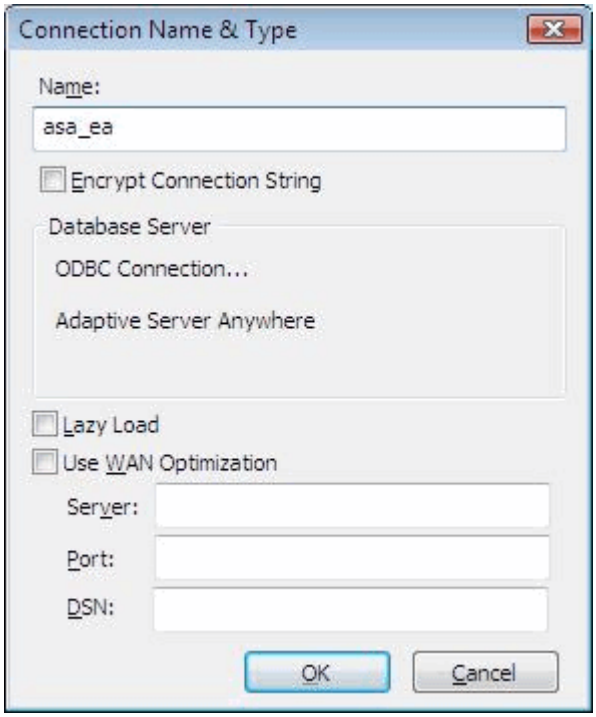
1. In the Open Project 8 dialog, select the **Connect to Server** checkbox or, on the Start Page, click on the **Connect to Server** link.
2. Click on the **[...]** (Browse) button, as you normally would to browse for a project. As you have selected the **Connect to Server** checkbox, the Data Link Properties dialog displays instead of the browse directories dialog.



3. Select **Microsoft OLE DB Provider for ODBC Drivers** from the list.
4. Click on the **Next** button. The Connection tab displays.



5. In the **Use data source name** field, click on the drop-down arrow and select the ODBC driver you set up to connect to your ASA repository.
6. Click on the **Test Connection** button to confirm that the details are correct.
7. If your test succeeded, click on the **OK** button.
8. If your test did not succeed, revise your settings.
9. After you have clicked on the **OK** button, the Connection Name & Type dialog displays.



10. Give the connection a suitable name so you can recognize it in the Recent Projects panel on the Open Project dialog 8.
11. If required, select the **Encrypt Connection String** checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
12. If required, select the **Lazy Load** checkbox to not load the full project view when the model is loaded. Instead, only the parts that are necessary to display the visible portion of the tree are loaded. This means that a model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.
13. If required, select the **Use WAN Optimization** 72 checkbox. (To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time.)

    If you select this checkbox, complete the next three fields (see your administrator for the correct values). Otherwise go to step 17.
14. In the **Server** field, type the network name or address of the optimizer server.
15. In the **Port** field, type the port on which the server is running on the remote machine.
16. In the **DSN** field, type the data source name of the database as it appears on the remote machine.
17. Click on the **OK** button to complete the configuration.

1.4.11.6 MSDE Server Data Repository

Follow the steps in Connect to an SQL Server Repository 52.

1.4.11.7 Progress OpenEdge Repository

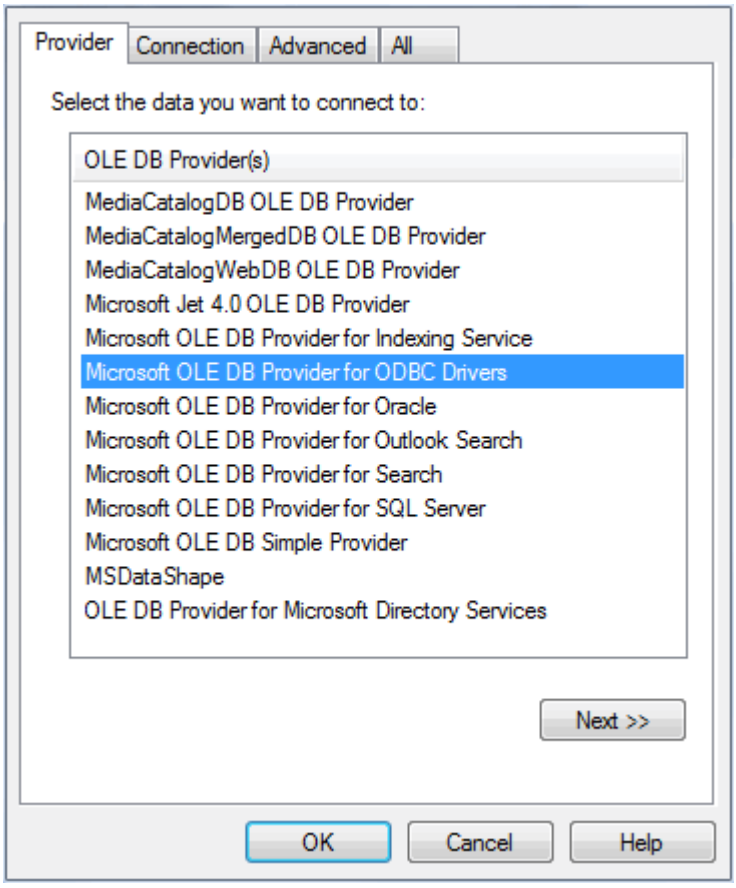
Note:

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

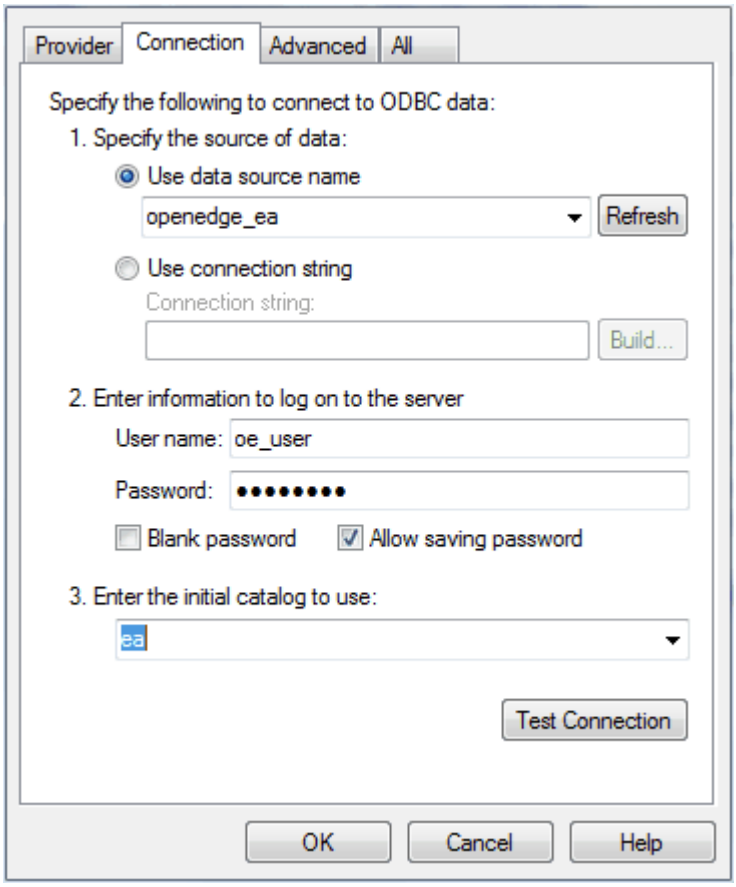
In order to use an OpenEdge data repository, you must connect to it in Enterprise Architect first; follow the steps below:

1. In the Open Project dialog 8, select the **Connect to Server** checkbox.

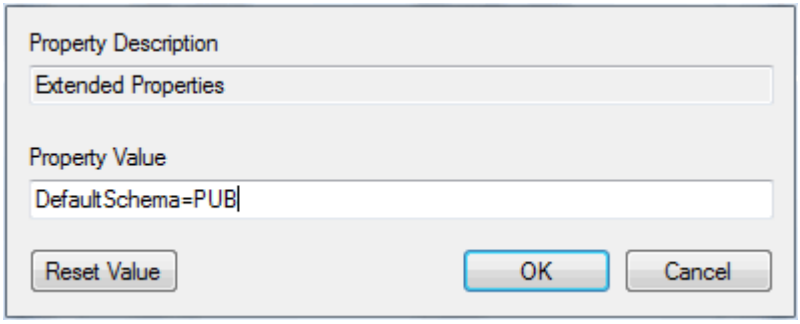
2. Click on the **[...]** (Browse) button, as you normally would to browse for a project. As you have selected **Connect to Server**, the Data Link Properties dialog displays instead of the Browse Directories dialog.



3. Select **Microsoft OLE DB Provider for ODBC Drivers** from the list.
4. Click on the **Next** button. The Connection tab displays.

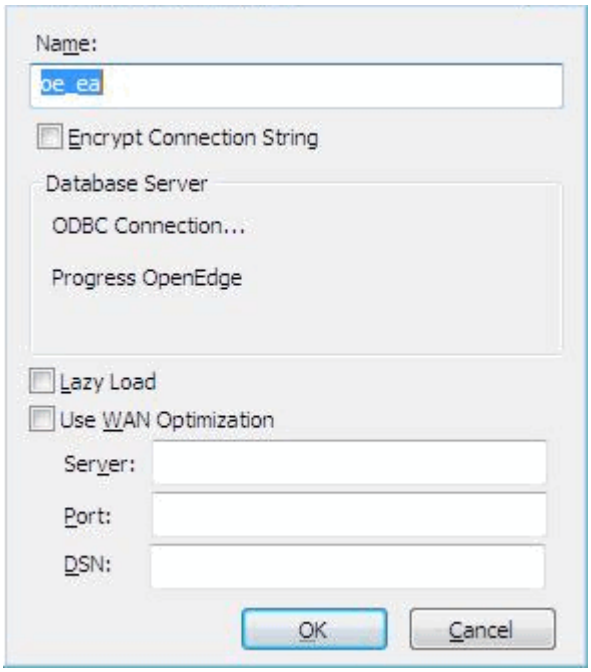


5. In the **Use data source name** field, click on the drop-down arrow and select the ODBC driver you have set up to connect to your OpenEdge repository. In the setup example 25 the driver title is **openedge_users**.
6. Enter the **User name** and **Password**.
7. Enter the initial catalog.
8. Click on the All tab, and double-click on **Extended Properties**.
9. In the **Property Value** field, edit the value to: **DefaultSchema=PUB**.



10. Click on the Connection tab again, and click on the **Test Connection** button to confirm that the details are correct.
11. If the test succeeds, click on the **OK** button. If the test does not succeed, revise your settings.
12. After you have clicked on the **OK** button, the Logon to Progress dialog displays.

13. Check the details, and click on the **OK** button. The Connection Name & Type dialog displays.



14. Give the connection a suitable name so you can recognize it in the Recent Projects panel on the Open Project dialog 8.
15. If required, select the **Encrypt Connection String** checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
16. If required, select the **Lazy Load** checkbox to not load the full project view when the model is loaded. Instead, only the parts that are necessary to display the visible portion of the tree are loaded. This means that a model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.
17. If required, select the **Use WAN Optimization** 72 checkbox. (To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time.)

    If you select this checkbox, complete the next three fields (see your administrator for the correct values). Otherwise go to step 21.
18. In the **Server** field, type the network name or address of the optimizer server.
19. In the **Port** field, type the port on which the server is running on the remote machine.
20. In the **DSN** field, type the data source name of the database as it appears on the remote machine.
21. Click on the **OK** button to complete the configuration.

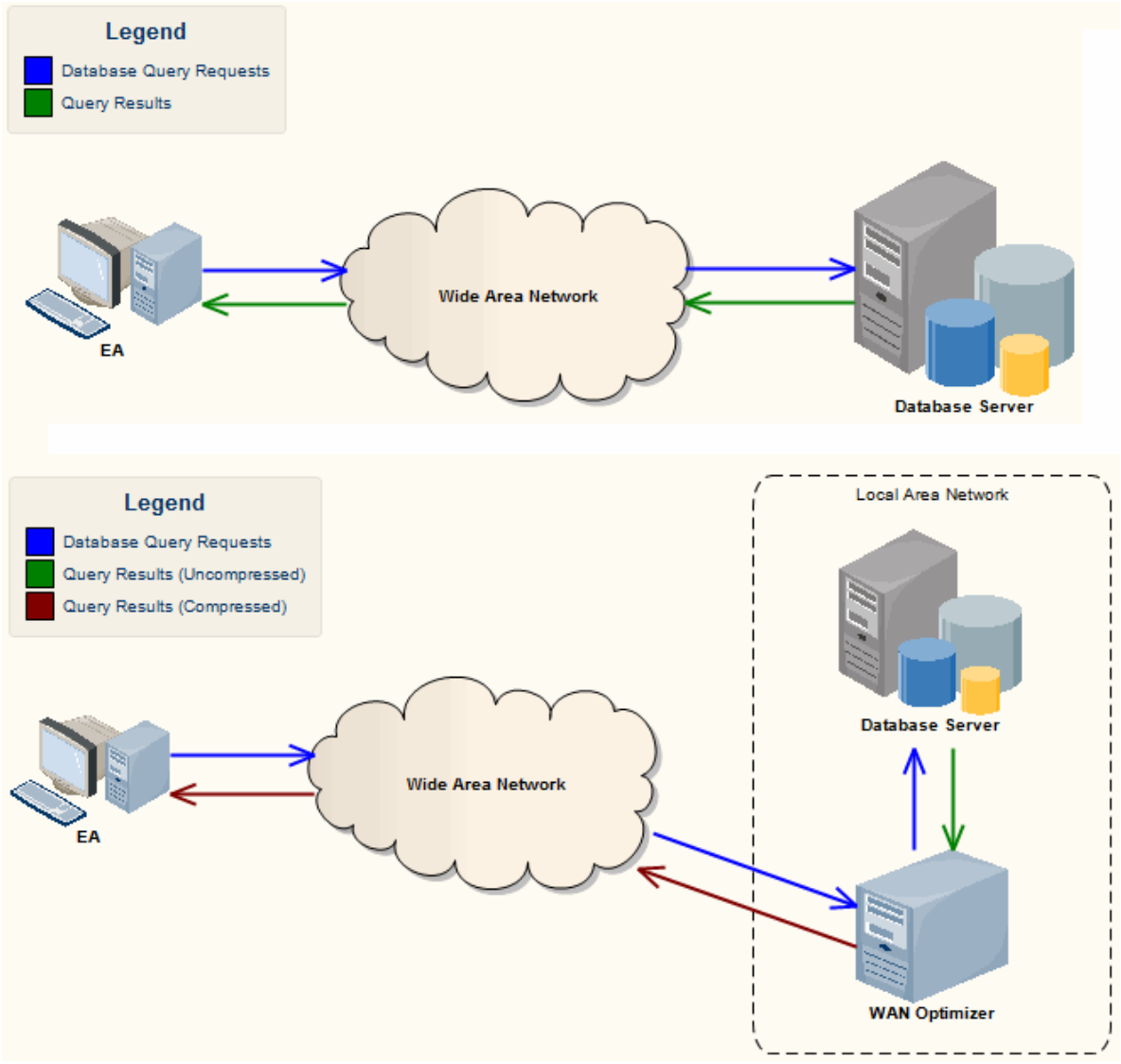
1.5 The WAN Optimizer

Introduction

The Sparx *Wide Area Network (WAN) Optimizer* is a lightweight server installed on a Local Area Network (LAN) connection to a Database Management System (DBMS) that hosts an Enterprise Architect repository. You can configure the server to listen for client connections on a particular port; it acts as a local proxy to execute queries and return the results in a compressed format to the client.

The WAN Optimizer significantly improves Enterprise Architect's performance in a WAN by reducing the amount of data transmitted and, in turn, the number of network calls made.

In the following diagram, transmission between Enterprise Architect and a DBMS is depicted first without and then with the WAN Optimizer.



You can download the WAN Optimizer installer from the Downloads page of the Registered Users section of the Sparx Systems website. The Wan Optimizer Service installer package provides two installable features for the target machine:

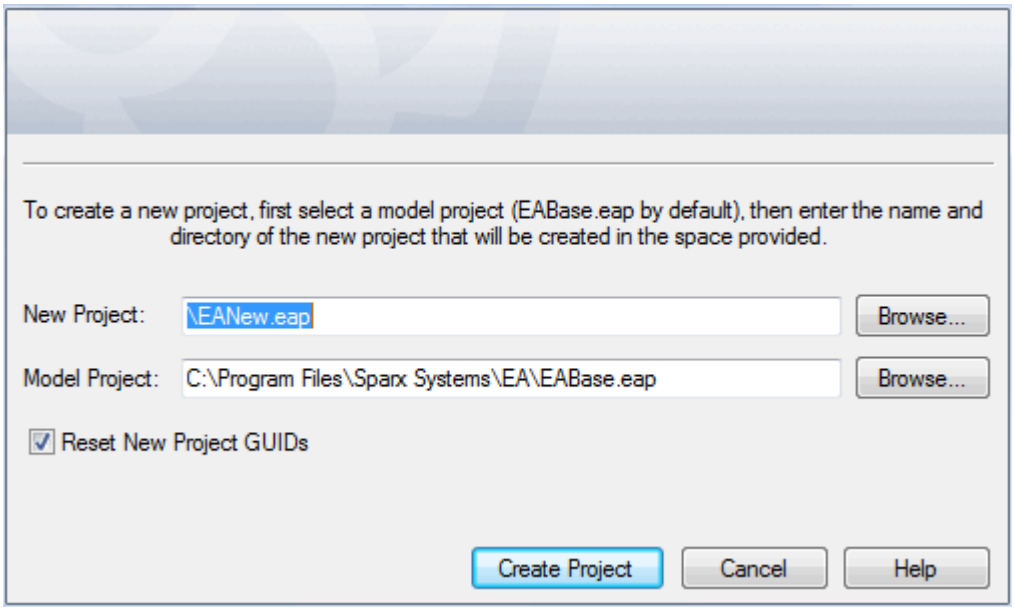
- WAN Optimizer Service - the installer also helps register and start the service on the target machine, and add it to the Windows Startup folder.
- WAN Optimizer Admin Client - to enable an administrator to administer and configure the service from a remote client.

The Optimizer has its own *Sparx WAN Optimizer User Guide*. See that User Guide for more information on:

- WAN Optimizer Components
- Installing and Starting the WAN Optimizer Service
- Configuring the Service
- Troubleshooting.

1.6 Copy a Base Project

To copy an existing Base Project, from the Start Page select the **Copy a Base Project..** option (see *Using Enterprise Architect - UML Modeling Tool*). The Create New Enterprise Architect Project dialog displays.



To create a new Enterprise Architect project, you must select a project template to form the base model for the new project. When you install Enterprise Architect a default model is installed called *EABase.eap.*

- To select the file path for saving your project, click on the **Browse** button after the **New Project** field. If this is to be a shared project, store the file on a shared network resource such as a Network Server or Workgroup Server.
- To replace all GUIDs in the source model with fresh GUIDs, select the **Reset New Projects GUIDs** checkbox.

Note:

If the new project is based on one that is already under version control, it is recommended that the **Reset New Projects GUIDs** checkbox be deselected. This prevents duplication of packages when the **Get Latest** facility is used.

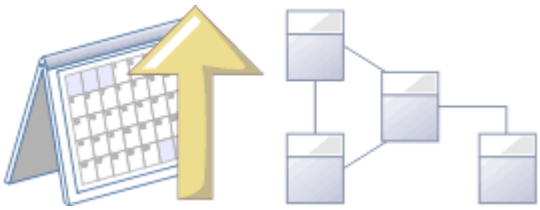
- To select the base model for your project 9, click on the **Browse** button after the **Model Project** field. The field defaults to *EABase.eap*; however you can select any existing model file (see the *Design a Custom Template* 9 topic).

When you have entered the filenames, click on the **Create Project** button to create your project. Click on the **Cancel** button to close the dialog without creating a new project.

Tip:

You can copy any Enterprise Architect project using Windows Explorer, and open the copied project as a new project.

2 Upgrade Models



The structure of Enterprise Architect project files is occasionally changed to support more features. When this happens, existing project files must be upgraded to the new format to ensure correct operation and to take advantage of all the new features.

When you try to load a project that was created in an early release of Enterprise Architect (for example, an archived project) using a recent release of Enterprise Architect, the system determines whether the project should be upgraded and, if the upgrade is necessary, displays the Upgrade Wizard 74, which takes you through the upgrade process.

Upgrading is a simple and quick process that brings your project to the current level to support all the latest Enterprise Architect features.

2.1 The Upgrade Wizard

When you first try to load an old project in a new version of Enterprise Architect, the system determines whether the project should be upgraded and, if the upgrade is necessary, displays the Upgrade Wizard.

The Upgrade Wizard:

- Advises you of the necessity to upgrade
- Advises you to back up the current project; it is essential to back up before any changes are made
- Checks which upgrade path is required
- Guides you through the steps to perform the upgrade
- Opens the newly converted project.

Notes:

- For replicated models: If the wizard detects the project you are opening is a replica and not a Design Master, a different upgrade path 75 is required.
- Once upgraded, the project cannot be opened with the version of Enterprise Architect in which it was created.

2.2 Upgrade Replicas

Models that have replication features added might have to be upgraded differently from regular projects.

If the model is a Design Master 101 (the root model of all other replicas) then you can upgrade the model to suit the current version of Enterprise Architect. After upgrading a Design Master you should re-create the replicas, rather than synchronizing.

If the model is not a Design Master, Enterprise Architect must first remove the replication features, then upgrade the project in the normal manner. The Upgrade Wizard 74 guides you through the steps.

3 Project Data Integrity

If you have a failed XMI import, network crash or other unforeseen event that could disrupt the integrity of information in the model, it is recommended to run the Project Integrity Check function 76. This examines all database records and ensures there are no 'orphaned' records or inaccurate or unset identifiers. You can run the integrity checker first in report mode to discover if anything should be corrected, and then run it again in repair mode.

When Enterprise Architect checks the model, it attempts to recover lost packages and elements, and generates a new package at the model root level called *_recovered_*. Check through any elements that are found and, if required, drag them into the model proper. If they are not required, delete them.

Note:

This function does NOT check UML conformance, only the data relationships and repository structure.

You can select a variety of items to check, and select either to just report on the state of your model, or to try and repair any inconsistencies. The recovery process tries to restore elements where possible, but in some cases simply deletes the lost records.

See Also

- Run SQL Patches 78

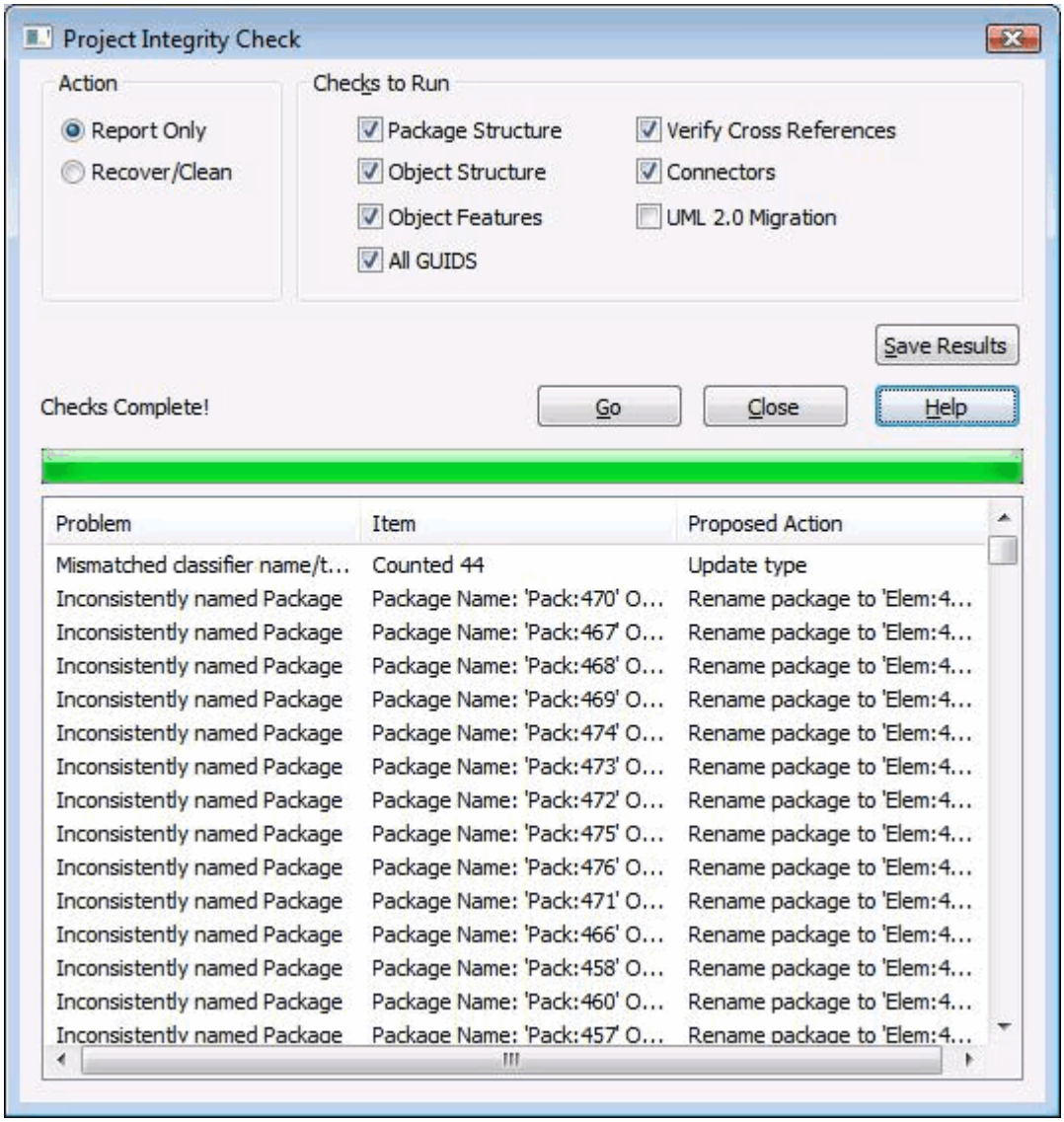
3.1 Check Project Data Integrity

Note:

In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Check Project Data Integrity** 183 permission to perform a data integrity check. See *User Security in UML Models*.

To check the data integrity of your project, follow the steps below:

1. Select the **Tools | Data Management | Project Integrity Check** menu option. The Project Integrity Check dialog displays.



2. Select the checks to run; the basic checks available are:
   - Package Structure
   - Object Structure
   - Object Features
   - All GUIDs
   - Cross References
   - Connectors
   - UML 2.0 Migration.
3. Select either:
   - the **Report Only** option to just view a report on the state of your model, or
   - the **Recover/Clean** option to attempt to recover and clean your project.

Warning:

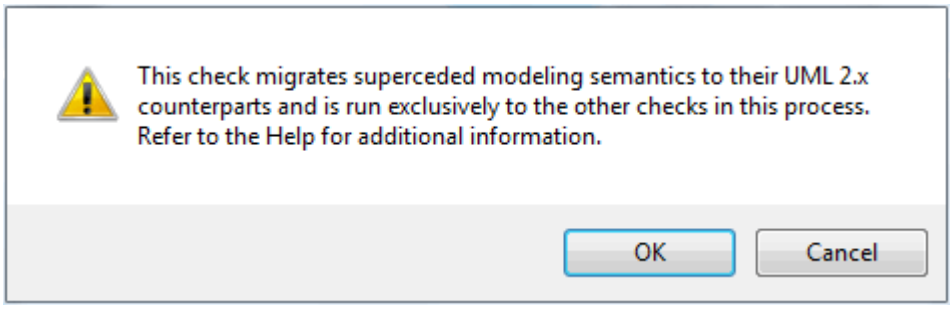
If you intend to select the **Recover/Clean** option, you should back up your project file first.

4. To write a log of the integrity check, click on the **Save Results** button and select a log file.
5. Click on the **Go** button to run the check. If you want to display the resulting information in a more readable layout, you can resize the dialog and its columns.

UML 2.0 Migration

The UML 2.0 Migration check enables you to migrate the project from UML 1.3 semantics to UML 2.0 semantics. The migration process currently converts activities that are invocations of operations into called operation actions as per the UML 2.0 specification. The UML 2.0 Migration option is an exclusive process that does not enable any of the other checks to be selected. To perform the UML 2.0 migration follow the steps below:

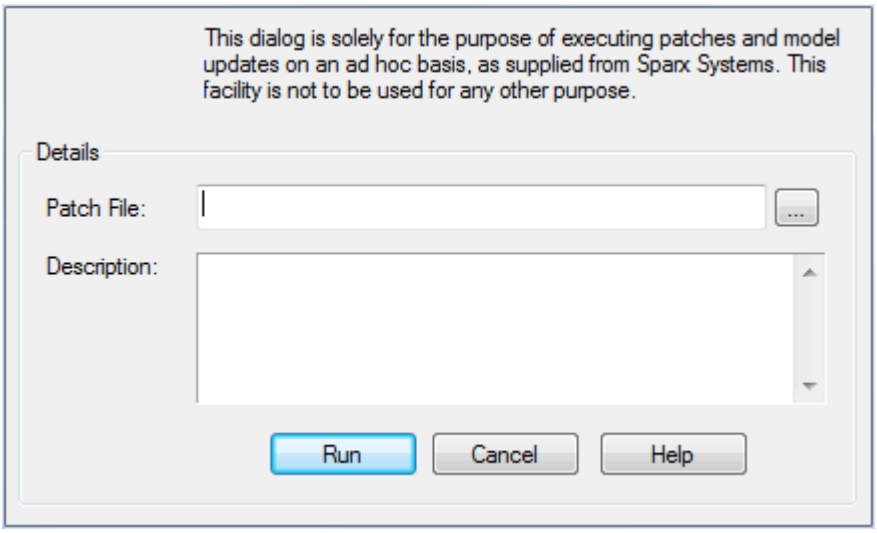
1. Select the **Tools | Data Management | Project Integrity Check** menu option. The Project Integrity Check dialog displays.
2. Select the **UML 2.0 Migration** checkbox and click on the **Go** button. The following message box displays:



3. To proceed, click on the **OK** button, or to cancel the migration click on the **Cancel** button.
4. If you are proceeding, click on the **Go** button on the Project Integrity Check dialog to perform the migration.

3.2 Run SQL Patches

Occasionally, Sparx Systems might release a patch to correct a model fault. To load such patches and run them, select the **Tools | Run Patch** menu option. The patch generally checks how many records are to be updated, and reports on what is to be done.



4 Project Data Transfer

The Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect support SQL Server 40, MySQL 38 and Oracle 9i, 10g and 11g 43 data repositories. At some point, it might become necessary to move a complete model from one repository to another, row by row, table by table.

The project data transfer function enables you to perform the following tasks:

- Upload an existing .EAP file to a SQL Server or MySQL repository
- Download a repository in MySQL or SQL Server to a .EAP file
- Move a repository from SQL Server to MySQL or from one server to another
- Move all records from a .EAP file with replication to a model with none (Remove Replication)
- Copy all records from a .EAP file to another (recommended after serious network crash or repeated database corruption)
- Copy all records from a JET 3.5 to JET 4 (Access 2000 or XP) repository - or back the other way.

See the *Perform a Project Data Transfer* 79 topic for instructions.

Note:

You cannot move a model from a source .EAP file of a version earlier than 3.5.0.

Warning:

All records in the target repository are overwritten.

4.1 Perform a Project Data Transfer

Warning:

During a project data transfer, all records in the target project are overwritten. Before performing the transfer, take a backup of the target project to ensure that you can recover any important information it contains.

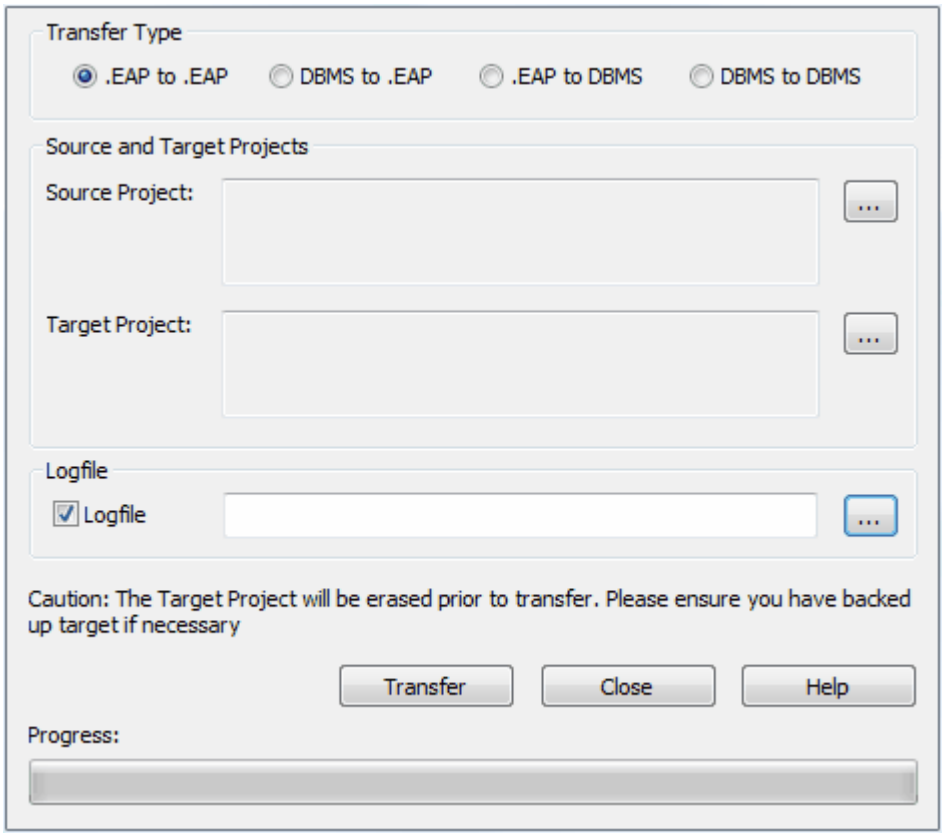
Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Transfer Data** 183 permission to transfer project data between repositories. See *User Security in UML Models.*
- You cannot move a model from a source .EAP file of a version earlier than 3.5.0.
- You must transfer data into a *repository* (where scripts have been run to set up tables), not just a database. If necessary, follow the steps below:
  - Install the DBMS software and create a database. Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic.
  - Run a script supplied by Sparx Systems (http://www.sparxsystems.com/resources/corporate/index.html#sql_scripts) to create the required tables.

To perform a project data transfer, follow the steps below:

1. Select the **Tools | Data Management | Project Transfer** menu option. The Project Transfer dialog

displays.



2. Click on the option for the required transfer type. You can choose from:
   - **.EAP to .EAP**
   - **DBMS to .EAP**
   - **.EAP to DBMS**
   - **DBMS to DBMS**
3. In the **Source Project** and **Target Project** fields, type or select the name or connection string for the Source and Target projects.
4. If you want to capture the transfer in a log file, select the **Logfile** checkbox and browse for the appropriate log file location.
5. Click on the **Transfer** button.

It is good practise to do a Project Compare 80 after this process to verify that all records are written.

4.2 Why Compare Projects?

It is sometimes useful to compare the size and row counts of two projects; for example, after a database crash, after import from XMI or after performing a deletion of model elements.

You can compare .EAP files to other .EAP files or to DBMS based repositories, or compare two DBMS repositories. Enterprise Architect examines the number of rows in each database and produces a report indicating the total records in each and the difference between the two. No examination is made of the data in the table, just the record count.

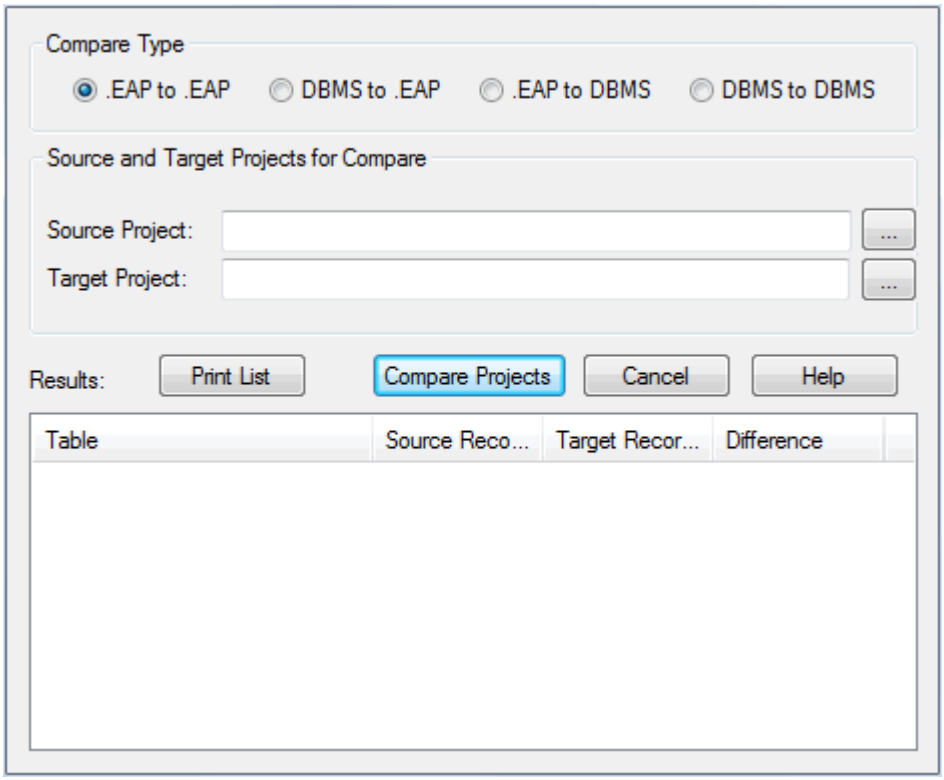
Comparing projects this way is a convenient 'sanity check' after restoring a backup or doing a project data transfer. If discrepancies are found, you must investigate further manually.

See the *Compare Projects* 81 topic for instructions.

4.3 Compare Projects

To compare projects, follow the steps below:

1. Select the **Tools | Data Management | Project Compare** menu option. The Project Compare dialog displays:



2. Select the option for the required comparison type. You can choose from:
   - **.EAP to .EAP**
   - **DBMS to .EAP**
   - **.EAP to DBMS**
   - **DBMS to DBMS**
3. In the **Source Project** and **Target Project** fields, type the name or connection string for the Source and Target projects.
4. Click on the **Compare Projects** button. The results of the comparison display in the panel at the bottom of the dialog.
5. If required, click on the **Print List** button to print the results.

4.4 Copy Packages Between Projects

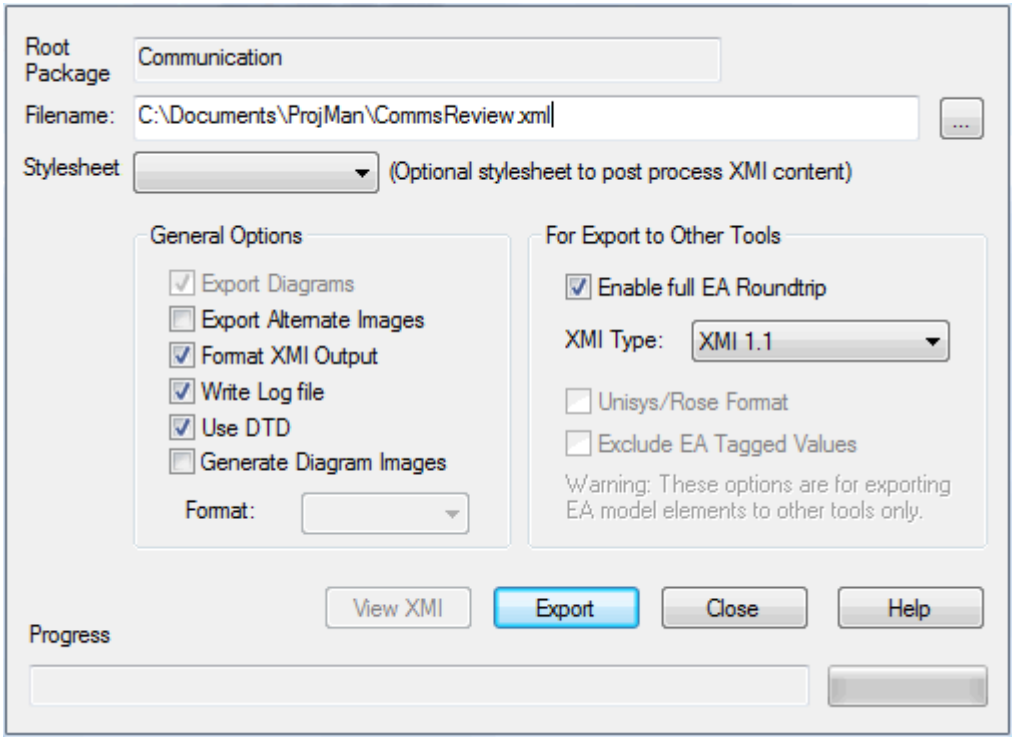
Using the XML import/export capabilities of Enterprise Architect, you can copy and move packages between Enterprise Architect projects. This gives you a high level of flexibility in building a project from re-usable parts and from elements produced in widely-dispersed geographic regions.

This procedure, with the **Strip GUID's** checkbox selected (see step 8, below) is effectively the same as copying packages within or between models. You would tend to use this export/import procedure for duplicating larger structures, such as complete models or projects, although exporting and importing individual child packages within the same model is just as feasible. You cannot export and import specific elements using this procedure, but the process of copying elements within or between packages, models and projects is derived from it. For further details of these copy procedures, see *UML Modeling With Enterprise Architect, UML Modeling Tool*.

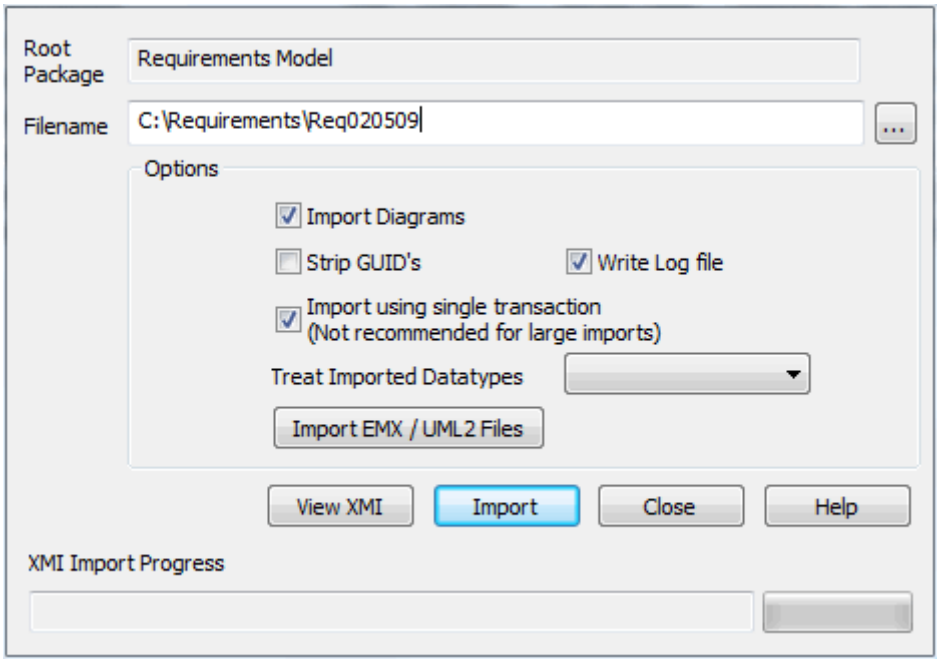
Procedure

To copy a package from one Enterprise Architect project to another, follow the steps below:

1. Open the Enterprise Architect project to copy from.
2. In the Project Browser, right-click on the package to copy. The context menu displays.
3. Select the **Import/Export | Export package to XMI file** menu option. The Export Package to XMI dialog displays.



4. Select the appropriate options and filename (see the *Export to XMI* 105 topic for further information).
5. Click on the **Export** button to begin the export process.
6. When the export is complete, open the recipient Enterprise Architect project. In the Project Browser, navigate to the location to import the package into.
7. Right-click to display the context menu, and select the **Import/Export | Import package from XMI file** menu option. The Import Package from XMI dialog displays.

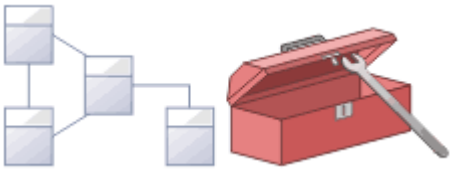


8. Select the appropriate options and filename (see the *Import from XMI* 106 topic for further information).
9. Click on the **Import** button. The package is copied from the source project to the destination project.

Note:

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 which case Enterprise Architect creates a replica of the original package.

5 Model Maintenance



This topic highlights some administrative functions you might have to carry out to maintain your model. Note that this maintenance applied to models created as .EAP files. The processes are not required for models stored in a DBMS:

5.1 Rename a Project

Important:

The only way to rename an Enterprise Architect project is at the Windows file system level.

To rename an Enterprise Architect project .EAP file, follow these steps.

1. If you have the project open, shut it down.
2. Ensure no other users have the file open.
3. Open Windows Explorer and navigate to the project.
4. Rename the project file using Windows Explorer.
5. You should keep the *.EAP* extension the same to preserve compatibility with the default project type, as installed in the registry at installation time.

5.2 Compact a Project

After some time, a project .EAP file might benefit from compacting to conserve space.

Notes:

- Compacting shuffles the contents of the model around, eliminating unused space and generally reducing the size of your model file.
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Administer Database** 183 permission to compact a project. See *User Security in UML Models.*

To compact a project, follow the steps below:

1. Ensure that no users have the target project open.
2. Select the **Tools | Manage .EAP File | Compact .EAP File** menu option.
3. Follow the on-screen instructions to complete the process.

Warning:

Always compact and repair projects on a local drive, never on a network drive.

5.3 Repair a Project

If a project has not been closed properly, such as during system or network outages, on rare occasions the .EAP file does not open correctly. In this case a message displays informing you the project is of an unknown database format or is not a database file.

Warning:

Never attempt to repair a project over a network connection; copy it to a local drive first.

Notes:

- Poor network connections can also cause this symptom.
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Administer Database** 183 permission to repair a project. See *User Security in UML Models.*

Repair a Project That Has Not Closed Correctly

To repair a project that was not closed properly, follow the steps below:

Note:

All users must be logged off the project you are attempting to repair.

1. Copy the project file to a local drive on your PC.
2. In Enterprise Architect, select the **Tools | Options** menu option and on the General page deselect the **Use Jet 4.0 - requires restart** checkbox.
3. Close and restart Enterprise Architect and open a place holder project to enable access to the **Repair .EAP File** facility.

   Note:

   This is NOT the project you intend to repair, it is a copy of it.
4. Select the **Tools | Manage .EAP File | Repair .EAP File** menu option.
5. Follow the on-screen instructions.

Ensure Integrity of the Repaired Project

An additional step you can use to ensure the integrity of your project is to use the **Remove Replication** feature.

1. Open Enterprise Architect, but when you are prompted to open a project, click on the **Cancel** button.
2. Select the **Tools | Manage .EAP File | Remove Replication** menu option.
3. Follow the prompts. When you are prompted for the Replica Project Browser window for your problem project, you might be given a warning about the project not being the *Design Master;* accept this warning. Click on the **Next** button.
4. Browse for the clean project (for example, *EABase.eap*). Click on the **Next** button.
5. Enter the path and name of the new project to be created, then click on the **Next** button.
6. Click on the **Run** button to run the removal process.
7. Once the removal process has been completed, open the project and do a check of the project contents. If the data is intact, backup the old project and replace it with the new version.

5.4 Manage Views

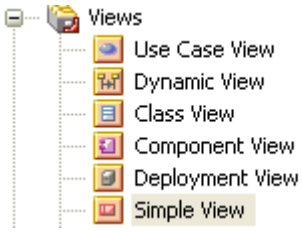
The top level packages in a model (below the project root nodes) can be created as *Views*. Views are used simply to subdivide the model into partitions such as Business Process, Logical View or Dynamic View. Unlike Model Packages, they do not have automatically-generated components and can be created only under a root node.

They are a good way to extend the model depending on specific requirements and modeling techniques.

There are 6 main types of View, each with their own package icon:

- Use Case View - containing, for example, Use Case diagrams, Analysis diagrams
- Dynamic View - containing, for example, Activity diagrams, Communication diagrams, Sequence diagrams, State diagrams
- Class View - containing, for example, Class Models, Code Engineering, Data Models
- Component View - containing, for example, Component diagrams
- Deployment View - containing, for example, Deployment diagrams
- Simple View.

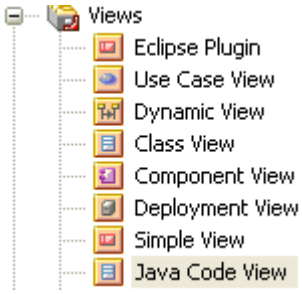
For information on these diagram types, see the *UML Dictionary*.



You can use the first five categories, or devise your own based on the Simple View. You can create 86 Views, rename 87 them, move them into a different order, or delete 88 them. Do this by right-clicking the mouse on the selected View to open the context menu, and choose the appropriate option.

5.4.1 Add Views

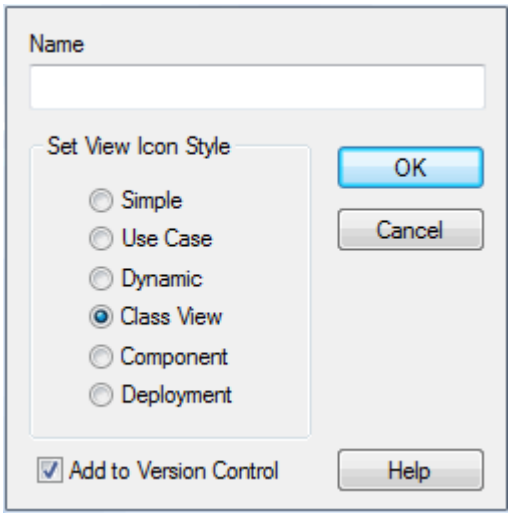
The example below shows an additional view called *Java Code View*, which has been appended to the end of the Views list.



Create a View

To create a View, follow the steps below:

1. Right-click on the model root node in the Project Browser. The context menu displays.
2. Select the **New View** menu option. The Create New View dialog displays.



3. In the **Name** field, type the name of the View.
4. In the Set View Icon Style panel, click on the radio button for the required View icon.
5. If the model root node is under version control 130, the **Add to Version Control** checkbox displays, defaulted to selected. If you do not want the new View to also be under version control, deselect the checkbox. (See *Version Control Within UML Models Using Enterprise Architect*.)
6. Click on the **OK** button.

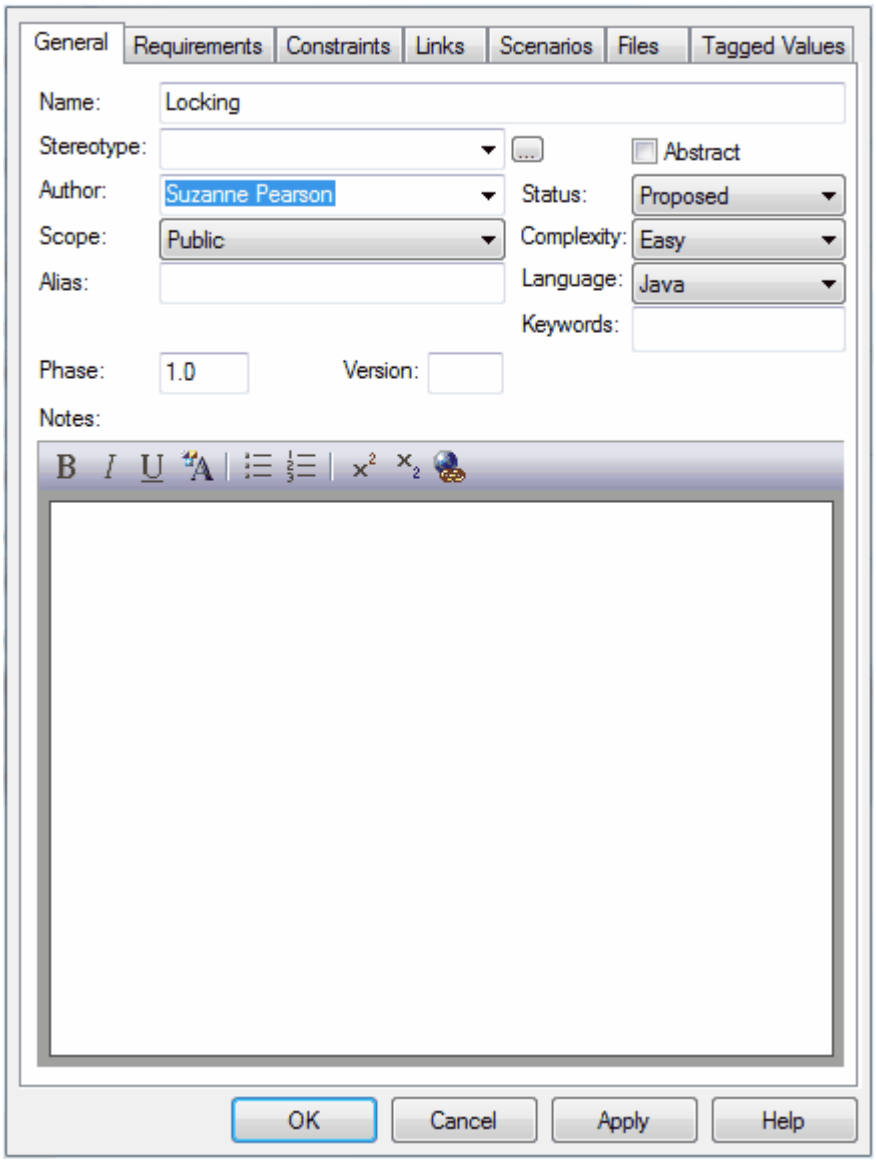
5.4.2 Rename Views

If required, you can rename a view.

Procedure

To rename a view, follow the steps below:

1. Right-click on the View in the Project Browser. The context menu displays.
2. Select the **Properties** menu option. The Package Properties dialog displays.



3. In the **Name** field, type the new name and click on the **OK** button.

5.4.3 Delete Views

If necessary, you can delete a view.

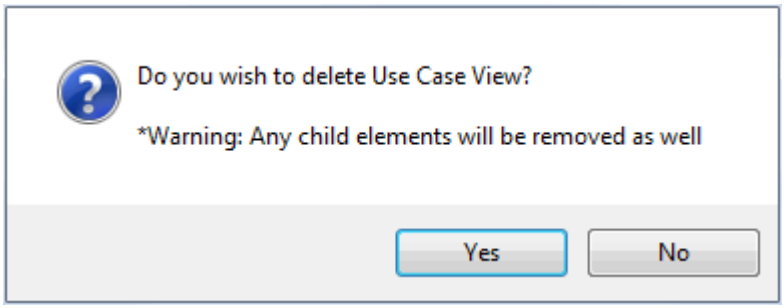
Warning:

If you delete a view, all its contents are deleted at the same time. It CANNOT be restored.

Procedure

To delete a view, follow the steps below:

1. In the Project Browser, right-click on the view to delete. The context menu displays.
2. Select the **Delete <viewname>** option. The following warning displays:



3. To delete the view and its contents, click on the **Yes** button. To cancel the deletion, click on the **No** button.

6 Model Validation

You use Model Validation to check UML models against known UML rules 92 (which you identify in configuring validation 92) as well as any constraints defined within the model, using the Object Constraint Language (OCL). You can run Model Validation against a single UML element, a diagram or an entire package.

Validating a UML:

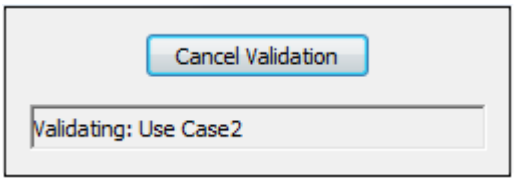
- **Element** validates the element and its children, features (attributes and operations) and relationships (connectors)
- **Diagram** validates the diagram itself (for correctness) as well as any elements and connectors within the diagram
- **Package** validates the package and all subpackages, elements, connectors and diagrams within it.

To use Model Validation, follow the steps below:

1. Select the package, diagram or element either from the Project Browser or within an open diagram.
2. Select the **Project | Model Validation | Validate Selected** menu option, or press **[Ctrl]**+**[Alt]**+**[V]**.

Enterprise Architect performs the validation, and displays the results in the Output window (see *Using Enterprise Architect - UML Modeling Tool*). (If the Output window does not automatically display, select the **View | System Output** menu option.)

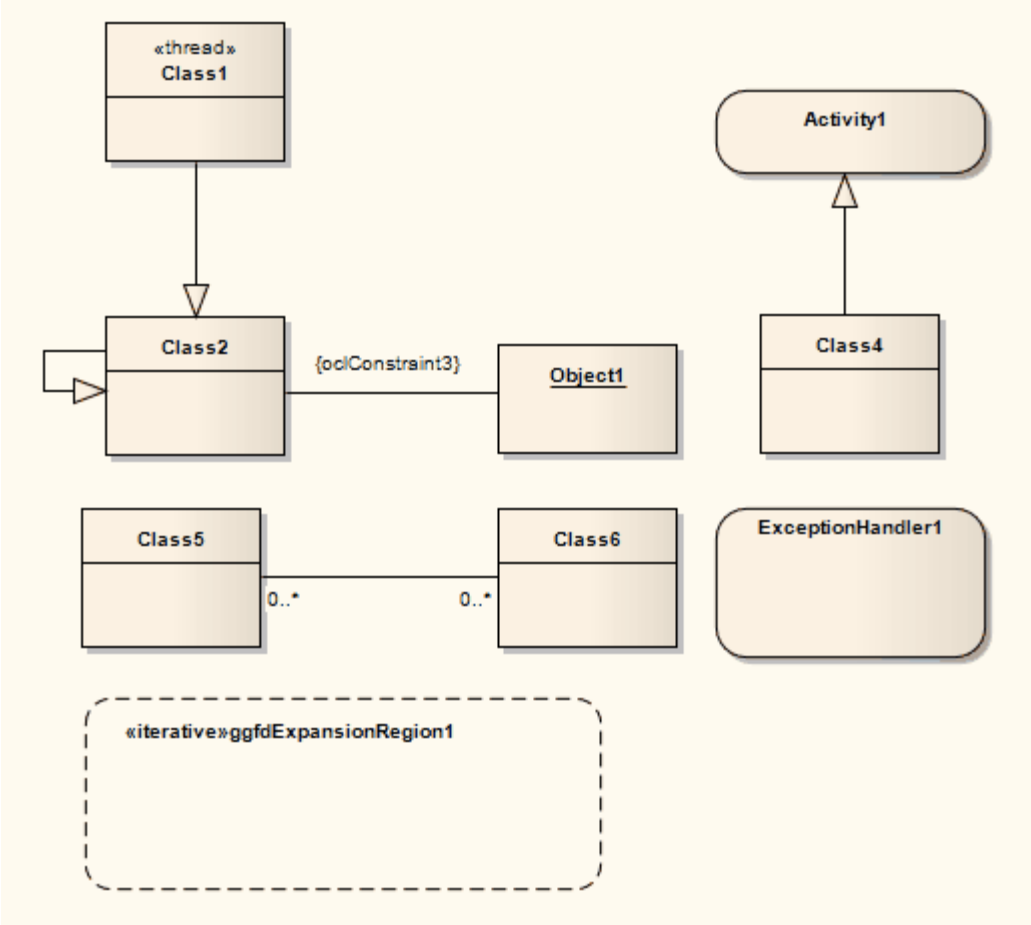
While performing the validation, Enterprise Architect also displays a progress window containing the **Cancel Validation** button, which enables you to cancel the validation process at any time.



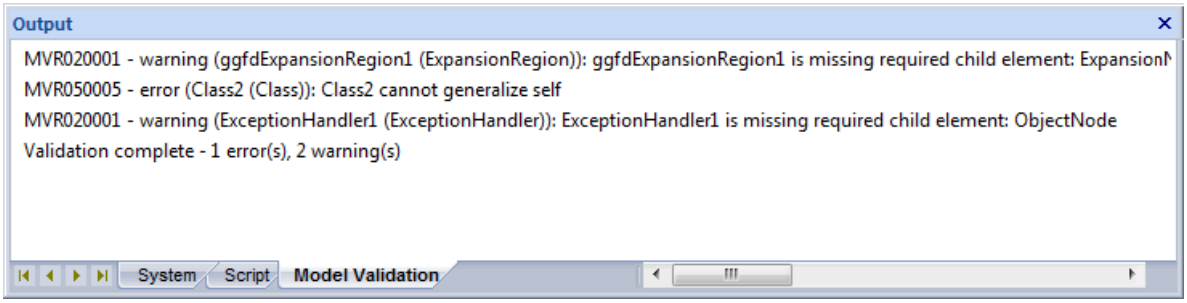
Alternatively, select the **Project | Model Validation | Cancel Validation** menu option.

Example Model Violations

The following UML diagram contains several basic violations.



If you run Model Validation on this diagram, Enterprise Architect displays the following violations in the Output window:



The validation results show that the diagram:

- Contains a UML ExpansionRegion (*ExpansionRegion1*) that is missing its child input ExpansionNode
- Contains an invalid self-generalization on *Class2* (UML elements cannot be self-generalized)
- Contains an OCL violation for the anonymous Association (between *Class2* and *Object1*)
- Contains a UML ExceptionHandler (*ExceptionHandler1*) that is missing its child input ObjectNode.

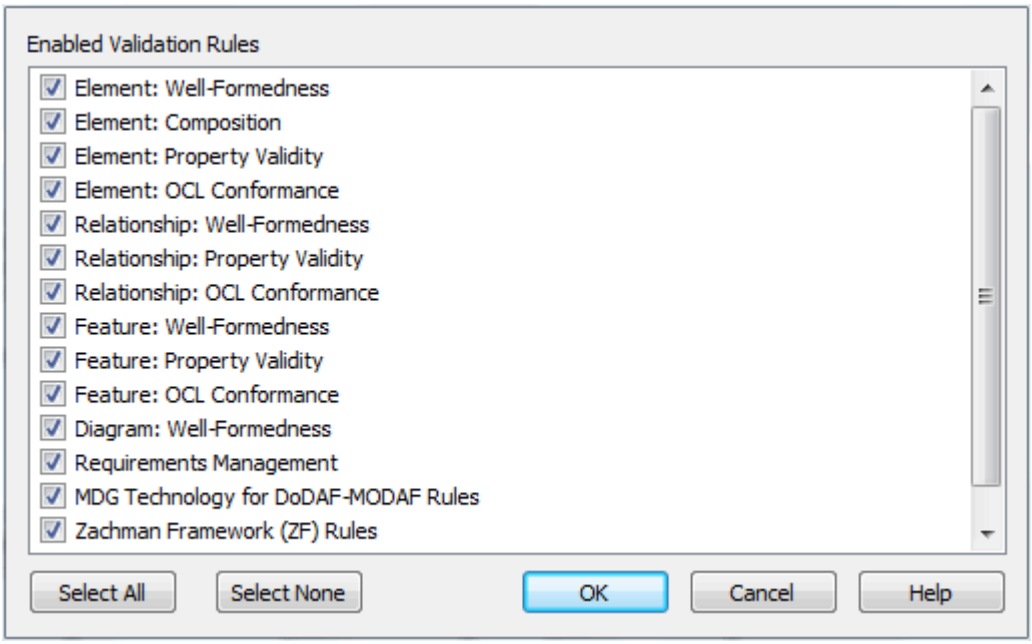
Note:

If you double-click on an error in the Output window, you select the diagram element that the error message refers to.

6.1 Configure Model Validation

Use the Model Validation Configuration dialog to enable and disable the rules 92 that are run with the model validator. You can define additional rules in this dialog from any additional Add-Ins that might be installed besides Enterprise Architect.

To display the Model Validation Configuration dialog, select the **Project | Model Validation | Configure** menu option.



Click on the checkbox against each Validation Rule to apply in performing a model validation 90.

Tip:

To disable UML syntax ("*The requested connection is not UML compliant*"), select the **Tools | Options** menu option, click on **Diagram** in the hierarchy, and in the General panel deselect the **Strict UML Syntax** checkbox.

When you perform a validation, each violation listed on the Output 91 window has a violation ID of the format *MVRxxnnnn*, where:

- *MVR* stands for Model Validation Rule
- *xx* is a hexadecimal number corresponding to the position of the validation rule in the Model Validation Configuration dialog, thus indicating which rule is applied and violated
- *nnnn* is the number of the violation message.

Therefore, messages with the ID *MVR01nnnn* indicate that the **Element: Well-Formedness** checkbox is selected and a violation of that rule has been detected. Messages with the ID *MVR0Annnn* indicate that the **Feature: OCL Conformance** checkbox (10th in order on the dialog, or Ath in hexadecimal) is selected and a violation of that rule has been detected.

6.2 Rules Reference

Model Validation works against a set of validation rules, arranged in the following groups:

- (Element, Relationship, Feature, Diagram): Well-Formedness 93
  Checks whether or not an element, relationship, feature or diagram is well-formed. This group of rules includes checks such as whether the item is a valid UML item and whether a diagram contains valid elements within it
- Element: Composition 93
  Checks whether or not a UML element contains valid children, whether it contains the right number of valid

children, and whether or not the element is missing any required children

- (Element, Relationship, Feature): Property Validity 94
  Checks whether or not the item has the correct UML properties defined, and whether the properties contain incorrect or conflicting values; for more information on these properties see the *Custom Properties* topic in *UML Modeling with Enterprise Architect – UML Modeling Tool*
- (Element, Relationship, Feature): OCL Conformance 94
  Validates an item against any defined constraints in OCL.

6.2.1 Well-Formedness

This group of rules checks whether or not an element, relationship, feature or diagram is well-formed. The rules includes checks such as whether the item is a valid UML item and whether a diagram contains valid elements within it. Reported violations include:

Violation ID	Description	Information
MVR010001	«Element» is not a valid UML Element	The element is not a recognized UML 2.1.1 element.
MVR050001	«Relationship» is not a valid UML Relationship	The relationship is not a recognized UML 2.1.1 relationship.
MVR050002	«Relationship» is not legal for «Start Element» --> «End Element»	The relationship between the given start and end elements is not valid for those elements.
MVR050003	«Parent Element»:isLeaf=true and cannot be generalized by «Child Element»	The Generalization relationship cannot exist between parent and child elements because the parent element is defined as a leaf element.
MVR050004	«Child Element»:isRoot=true and cannot generalize «Parent Element»	The Generalization relationship cannot exist between parent and child elements because the child element is defined as a root element.
MVR050005	«Element» cannot generalize self	The element cannot be self-generalized.
MVR0B0001	Statechart violation: «extended information»	The State diagram contains a UML violation; see the extended information for more information about the detected violation.

6.2.2 Element Composition

This group of rules checks whether or not a UML element contains valid children, whether it contains the right number of valid children, and whether or not the element is missing any required children.

Error ID	Description	Information
MVR020001	«Element» is missing required child element «Child Element».	The element is missing a child element of type *Child Element*.
MVR020002	Invalid UML package child.	The element cannot be a direct package child and must be a child of another element (for example: Ports must be children of other elements, and not direct UML package members).
MVR020003	Invalid child «Child Element name» («Child Element Type»).	The child element is invalid on the tested parent element.

6.2.3 Property Validity

This group checks whether or not an element, relationship or feature has the correct UML properties defined for it and whether they contain incorrect or conflicting values. For more information about these properties see the *Custom Properties* topic in *UML Modeling with Enterprise Architect – UML Modeling Tool*.

Error ID	Description	Information
MVR030001	«Element»:«Property» property is undefined	The element property contains no value.
MVR030002	«Element»:«Property» property has invalid value: "«Value»"	The element property contains an invalid value.
MVR030003	«Element»:isLeaf=true and cannot be abstract	The element's *isLeaf* and *isAbstract* properties are both set to **true**, which is invalid.
MVR060001	«Relationship»:«Property» property is undefined	The relationship property contains no value.
MVR060002	«Relationship»:«Property» property has invalid value: " «Value»"	The relationship property contains an invalid value.
MVR090001	Attribute/AssociationEnd mismatch, «Attribute»: «Mismatch description»,...	The given attribute has an *associationEnd* of the same name but they differ in the listed details.

6.2.4 OCL Conformance

This group validates an element, relationship or attribute against any defined constraints in the Object Constraint Language (OCL). OCL is used to describe expressions on UML models, and to express side-effect free constraints. You can add OCL constraints to any element, relationship or attribute in Enterprise Architect.

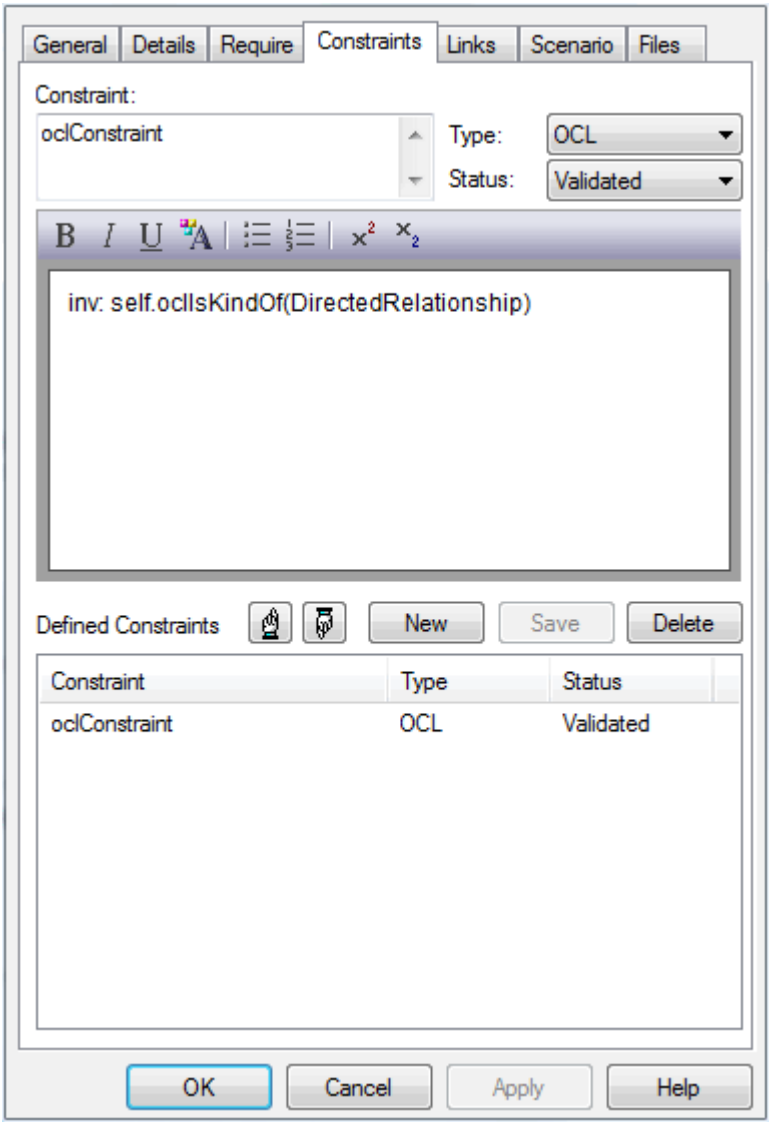
Error ID	Description	Information
MVR040001	OCL violation: «violated OCL»	The element violates the OCL constraint specified.
MVR070001	OCL violation: «violated OCL»	The relationship violates the OCL constraint specified.
MVR0A0001	OCL violation: «violated OCL»	The attribute violates the OCL constraint specified.

Important:

To have a valid OCL constraint, the syntax must be correctly formed. If the expression is not correct, Enterprise Architect displays a message stating that the OCL constraint is not valid.

Define OCL Constraints for an Element

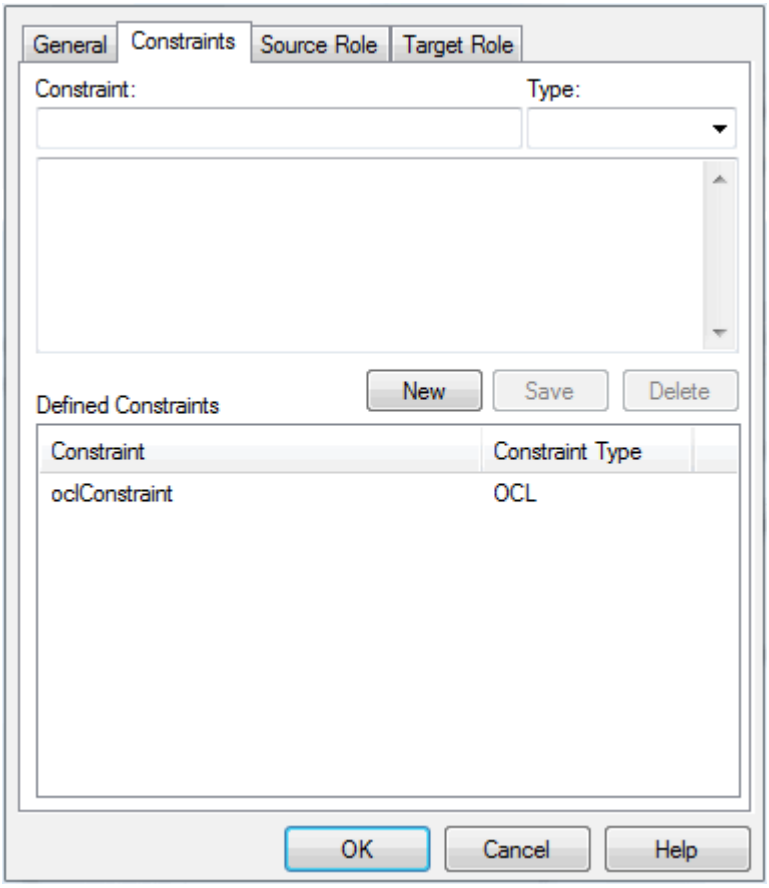
You can add OCL constraints to an element using the Properties dialog (**Element | Properties**). Select the Constraints tab, click on the **Type** drop-down arrow and select **OCL**.



To perform an OCL Validation, display the Model Validation Configuration 92 dialog and select the **Element: (OCL) Conformance** checkbox. Any OCL violations are recorded in the Model Validation Output window 91.

Define OCL Constraints for a Relationship

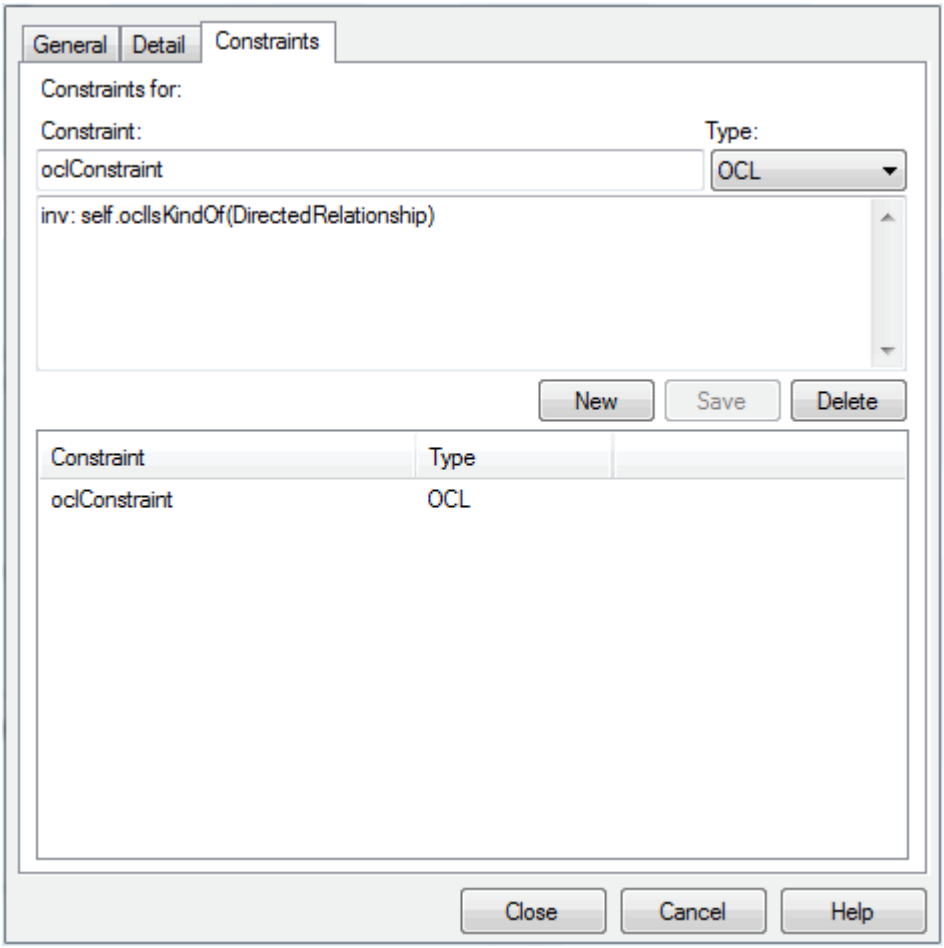
You can add OCL constraints to a relationship using the Properties dialog (right-click and select the **<type> Properties** context menu option). Select the Constraints tab, click on the **Type** drop-down arrow and select **OCL**.



To perform an OCL Validation, display the Model Validation Configuration dialog and select the **Relationship: (OCL) Conformance** checkbox. Any OCL violations are recorded in the Model Validation Output window.

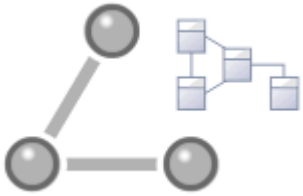
Define OCL Constraints for an Attribute

You can add OCL constraints to a feature using the Properties dialog (**Element | Attributes**). Select the Constraints tab, click on the **Type** drop-down arrow and select **OCL**.



To perform an OCL Validation, display the Model Validation Configuration dialog and select the **Feature: (OCL) Conformance** checkbox. Any OCL violations are recorded in the Model Validation Output window.

7 Model Sharing and Team Deployment



Introducing Team Development

Enterprise Architect offers a diverse set of functionality designed specifically for sharing projects 99 in team-based and distributed development 99 environments. Project sharing can be achieved through network deployment of model repositories, replication 100, XMI Import/Export 104, Version Control 130, Package Control 110 and User Security 172.

Network deployment can be undertaken using two different schemas for deployment, using either:

- .EAP based repositories or
- DBMS server based repositories.

Replication requires the use of .EAP based repositories, and cannot be performed on repositories stored on a DBMS server. DBMS server based repositories offer better response times than .EAP files on networks due to the inherent structure of the DBMS. DBMS also offers a better solution when networking problems are encountered, as they have the ability to backtrack transactions caused by external breakdowns.

Replication

Replication is a simple process that enables data interchange between .EAP based repositories (not DBMS) and is suitable for use in situations where many different users work independently. Modelers merge their changes into a Design Master only as required. It is recommended that a backup is carried out prior to replication.

XMI Import Export

XMI Import/Export can be used to model discrete packages that can be exported and shared between developers. XMI enables the export of packages into XML files which can then be imported into any model.

Package control can be used to set up packages for version control and to enable batch export of packages using XMI. Version Control enables a repository to be maintained by a third-party source code control application that is used to control access and record revisions.

Security

User security is used to limit the update access to model elements. It provides control over who in a project can make changes to model elements. User Security is documented in *User Security in UML Models*.

Further Information

For more information regarding the use of Enterprise Architect with shared models and team deployment please see the *Deployment of Enterprise Architect* white paper available from: www.sparxsystems.com/downloads/whitepapers/EA_Deployment.pdf.

Note:

DBMS Repository support and User Security are available with the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect.

7.1 Share Enterprise Architect Projects

Note:

Project Sharing and Replication are only enabled in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect.

Sharing a project among a team of designers, developers and analysts is the most efficient way of using Enterprise Architect to manage a team development. Many people can work on the model at the same time and contribute their particular skill. Team members can always see what the latest changes are, keeping the team informed and up to date with the project status.

You can share an Enterprise Architect project in three ways:

1. Using a shared network directory 99. In this scenario you place the project file on a shared network drive. Individual developers and analysts can then open and work on the project concurrently. Note that some project views (especially the Project Browser) require occasional refreshing to see changes made by other users.
2. Using replication. Replication 100 is a powerful means of sharing projects between isolated or mobile users. In the replication scenario a project is converted to a design master, then replicas made of the master. Users take the replicas away, modify the project, then bring their replicas back to be synchronized with the master file.
3. Using a shared DBMS-based repository (Corporate, Business and Software Engineering, System Engineering and Ultimate editions).

7.2 Share Projects on Network Drive

The easiest way to share a project amongst a work group of developers and analysts is to place the project file on a shared network drive and have people connect concurrently from their Workstation.

Note:

Enterprise Architect accepts a number of concurrent connections without issue, although there can be occasional 'lock-outs' when one user tries to access or update something another user is in the process of modifying.

Network Issues

The main issues with shared network access are:

- Changes to the Project Browser are not automatically updated. To compensate for this, users must occasionally reload 168 their project to view any project changes at this level (See *Version Control Within UML Models Using Enterprise Architect*).
- If two or more people work on the same diagram concurrently, unexpected results can occur. It is best to enable only one analyst to work on a diagram at a time.
- If a user's machine crashes, the network suffers an outage or a machine is turned off unexpectedly, the project file might require repair to compensate for the sudden inconsistency. A repair 85 facility is provided (select the **Tools | Manage .EAP File | Repair .EAP File** menu option) to carry out this task. This only applies to the file-based version of Enterprise Architect; the DBMS-based version does not suffer this problem.

7.3 Distributed Development

Enterprise Architect supports distributed development using two different techniques, as described below.

Replication

Use the Replication features to enable geographically separated analysts to update and modify parts of the model in replicas, then merge these back together at a central location. For further information see the *Replication* 100 topic.

XMI Import/Export

Use the XMI-based Import/Export facility to model discrete packages, export to XML and share among the development team. This approach has several benefits over replication:

1. You can assemble a model from only the parts necessary to get your job done.
2. You can assemble a full model if required.
3. You can assemble a model from different package versions for different purposes (such as customer visible, internal release only).
4. You can roll-back parts of a model as required.
5. There is less chance of 'collisions' between developers if each works on a discrete package.
6. The process is controllable using a version control 130 system. See *Version Control Within UML Models Using Enterprise Architect.*

Use the **Import/Export** menu options (below) to access this feature; they are available through the **Project | Import/Export** submenu. Also see the *XMI* 104 topic for further information on XMI-based import and export.

The Controlled Package 110 feature can also be used to assist in the process.

Note:

XMI based import/export is UML1.3 / XMI1.1 compliant. You can also write XML based tools to manipulate and extract information from XML files to enhance the development process.

7.4 Replication

In addition to sharing Enterprise Architect projects in real time over a network, you can also share projects using Replication, options for which are available through the **Tools | Manage .EAP File** menu.

Replication enables different users to work independently of one another, and to merge their changes at a later time. To avoid difficulties in this inevitably hazardous process, please read all sections of this topic carefully.

Enterprise Architect Merge Rules

Enterprise Architect follows these rules in merging:

- Additions are cumulative; that is, two replicas each creating three new Classes result in six new Classes after merging.
- Deletions prevail over modifications; if one replica changes a Class name and other deletes the Class, performing a merge results in both files losing the Class.

Conflicting modifications appear in the Resolve Replication Conflicts dialog (**Tools | Manage EAP File | Resolve Replication Conflicts** menu option). See Resolve Conflicts 103 for details on how to deal with conflicting modifications.

Use Replication

To use replication, follow the steps below:

1. Convert the base project to a design master 101 using the **Tools | Manage .EAP File | Make Design Master** menu option.

2. Create replicas 101 from the design master using the **Tools | Manage .EAP File | Create New Replica** menu option.
3. Take the replica away and work on it as required, then bring it back for synchronization with the design master.
4. Synchronize the replicas 102. During synchronization, all changes to both the master and the replica are propagated in both directions, so at the end they both contain the same information.

Upgrades and Replicas

When you upgrade your version of Enterprise Architect, you must not open a replica until you have opened the design master and then synchronized the replicas with the master. You cannot directly upgrade a replica 102.

Avoid Change Collisions

If two or more people make changes to the same element - for example, a Class - Enterprise Architect arbitrarily overwrites one person's change with another's. To avoid this, different users should work on different packages.

However, since Enterprise Architect does not enforce this rule, it is possible for users' work to conflict. To minimize the difficulties this causes, please note the following guidelines:

- If users are likely to have worked in the same area of the model, they should both witness the synchronization and confirm that they are happy with the net result.
- If small pieces of information have been lost, they should be typed into one of the merged models after synchronization.
- If a large piece of information has been lost (for example, a large Class note that was overwritten by another user who had made a minor change to the same Class) use the Resolve Replication Conflicts 103 dialog.

Disable or Remove Replication Features

If you have converted a project to a design master but now want to disable the replication 102 features, use the **Tools | Manage .EAP File | Remove Replication** menu option. Make sure you back up all your files first!

7.4.1 Design Masters

A *design master* is the first converted Enterprise Architect project that supports replication. From the design master you create replicas that can be modified independently of the master project and re-merged later.

Create a Design Master

To create a design master, follow the steps below:

1. Take a back-up of the required Enterprise Architect project.
2. Select the project in the Project Browser.
3. Select the **Tools | Manage .EAP File | Make Design Master** menu option and follow the on-screen instructions.

7.4.2 Create Replicas

Note:

In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Replicas** 183 permission to create a replica. See *User Security in UML Models.*

To create a replica, follow the steps below:

1. First create a design master 101, then select the **Tools | Manage .EAP File | Create New Replica** menu option and follow the on-screen instructions.
2. This process creates a replica of the current project which can then be modified independently, and afterwards re-merged with the main project.

7.4.3 Synchronize Replicas

To copy changes from one member of the replica set to another, use the **Synchronize Replicas** menu option. Note that information is copied both ways, including deletes, updates and inserts.

Note:

In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Replicas** 183 permission to synchronize a replica. See *User Security in UML Models.*

Synchronize a Replica

To synchronize a replica and a design master, follow the steps below:

1. Open the design master project file.
2. Select the **Tools | Manage .EAP File | Synchronize Replicas** menu option.
3. Locate and select the required replica to merge the open project and the replica.

Note:

When you synchronize, both projects end up containing identical information.

Change Collisions

Note that if two or more people work on the same element (or package or diagram) then the replication engine has problems in resolving which change is the master. To avoid this, always work on separate areas in the model when you are using replicas. You can also use the **Tools | Manage .EAP File | Resolve Replication Conflicts** 103 menu option.

7.4.4 Remove Replication

Replication makes many changes to the database structure of your model. As a consequence the model file becomes considerably larger with additional information. If you no longer require a model to be replicable, you can remove all replication features.

Remove Replication

To remove replication, follow the steps below:

1. If a repository is not open, the menu option for removing replication is not enabled. A temporary repository (not the one having replication removed) must be open at the time. Ensure you have a repository open at the time of creation.
2. Select the **Tools | Manage .EAP File | Remove Replication** menu option, to open the Remove Replication Wizard.
3. Enter the full path and file name of the project to have replication removed. Click on the **Next** button.
4. Enter the full path and file name of the base Enterprise Architect model (with no replication) to act as template. Click on the **Next** button.
5. Enter the full path and required file name for the output file. Click on the **Next** button.
6. Select whether to have a log file created, and enter a file name for the log file.
7. Click on the **Run** button to begin removing replication. Enterprise Architect creates a new project containing all the model information.

Your model has now had replication removed, and should be considerably smaller.

7.4.5 Upgrade Replicas

With new releases of Enterprise Architect there could be changes to the underlying project structure, such as more tables or changed queries. If you are using replicas to share and work with Enterprise Architect projects, it is very important that you open the design master 101 before opening any of the replicas with an updated version of Enterprise Architect.

Warning:

Upgrading Replicas takes special care!

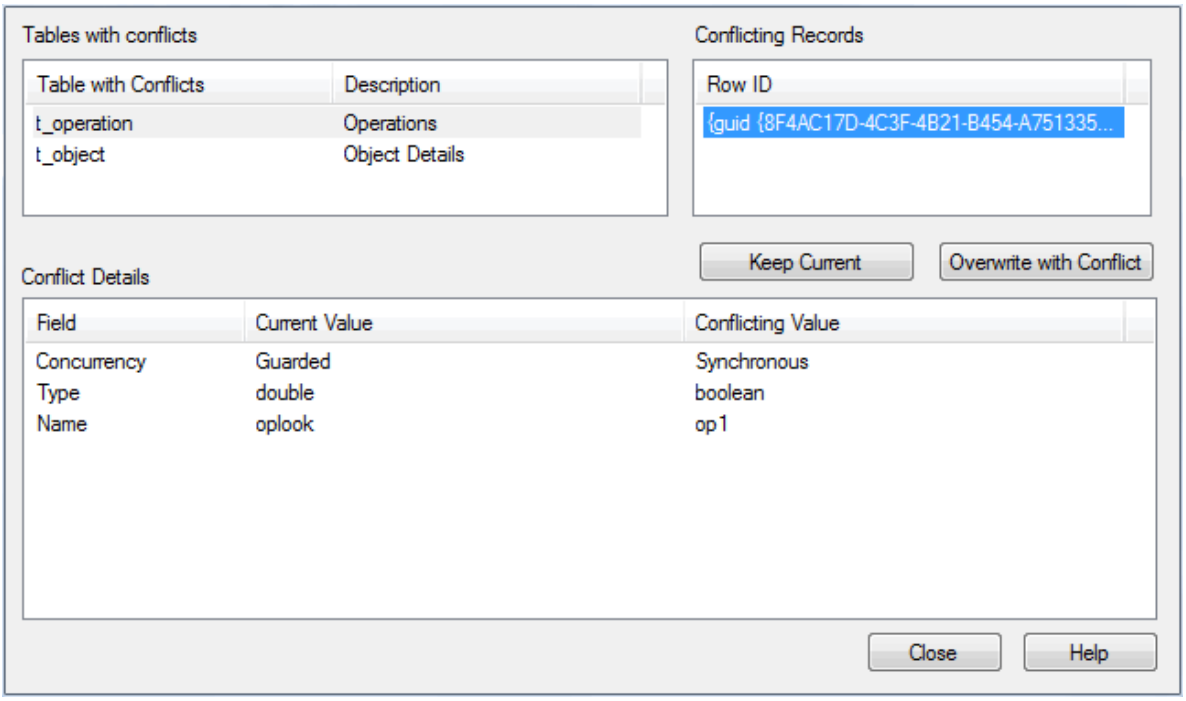
Changes to the database design in a replica set can ONLY be done to the design master. Next time the replicas are synchronized 102 with the master, the design changes are propagated through to the replicas. Trying to update a replica first at best does nothing, and at worst causes the update of the master to fail.

One other strategy is to remove 102 replication from a copy of the replica set, upgrade that project and convert it into a new design master from which new replicas are created.

7.4.6 Resolve Conflicts

When two or more people have changed the same element between synchronization points, Enterprise Architect has trouble resolving which change to accept and which to discard. The choice is made based on some rules within the JET replication manager, but the discarded changes are also stored so you can manually override that choice.

After synchronizing replicas, open the Resolve Conflicts dialog and check that there were any conflicts. Select whether to accept each change or use one of the discarded changes instead.



Recommendations for Resolving Conflicts

Enterprise Architect stores model information in database records. When two records have been modified in different ways by different users, they appear in this dialog.

Normally it is not necessary or desirable to examine conflicts, since they represent relatively inconsequential pieces of information that can very easily be modified through the normal Enterprise Architect interface; for example, by moving a diagram element.

The only case in which this dialog should be used is where a substantial piece of information has been overridden by another user, and you want to retrieve it. Follow the steps below:

1. In the Table with Conflicts list, click on the entry that is likely to contain the lost information.
2. Click on each entry in the Conflicting Records list.
3. When the lost information appears in the Conflict Details list, click on the **Overwrite with Conflict** button.

8 XMI Import and Export

What is XMI?

XML Metadata Interchange (XMI) is an open standard file format that enables the interchange of model information between models and tools. XMI is based on XML, and is defined by the OMG. 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.

Enterprise Architect supports the XMI 1.1, 1.2 and 2.1 specifications, but 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. XMI 1.1 has support for UML 1.3, whereas XMI 2.1 has support for UML 2.0 and UML 2.1.

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. This greatly improves the flexibility and robustness of Enterprise Architect models by enabling Analysts and Modelers to externalize model elements in XMI for version control, distributed development, post processing and transferring packages between models. When performing Enterprise Architect-to-Enterprise Architect transfers, ensure that the XMI version selected is 1.1 or 2.1.

XMI Tasks

Tasks you might perform in importing and exporting XMI include:

- Setting XML Options: XMI import, export and package control all rely on saving and loading XML files; you can set a number of options to streamline this process (see the *Defaults and User Settings* topic in *Using Enterprise Architect - UML Modeling Tool*)
- Exporting a package 105 to XMI in XMI 2.1 (and earlier)
- Importing from XMI 106 with support for XMI 2.1 (and earlier)
- Setting up controlled packages 110
- Manually controlling a package 116 by linking it to an XMI file
- Batch exporting 115 controlled packages
- Batch importing 106 controlled packages
- Factoring in the limitations of XMI 109
- Applying a UML Data Type Definition 109 (DTD)

For further information on XMI, including specifications, see the OMG *XML/XMI Technology* topic.

Notes:

- XMI 2.1 exported by Enterprise Architect 7.0 (or later) might not be correctly imported into earlier versions of Enterprise Architect.
- 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.

  However, an error does occur if you are importing an XMI 1.1 file that has been exported with the UML_EA.DTD file, and the UML_EA.DTD file is not present in the same directory as the XMI file.

Important:

When you import an XML file over an existing package, ALL information in the current package is deleted first. Before you import the XML file, please make sure you do not have important changes that you do not want to lose.

8.1 Export to XMI

You can export a package to an XMI (XML based) file. This enables you to move Enterprise Architect Model elements between models, for distributed development 99, manual version control 116 and other benefits. It also enables limited export of Enterprise Architect model elements to Rational Rose and other tools that implement the UML 2.1 XMI 2.1 standard, the UML1.4 XMI 1.2 standard, or the UML 1.3 XMI 1.1 / XMI 1.0 standard.

For more information regarding the limitations of XMI exporting read the *Limitations of XMI* 109 topic.

Notes:

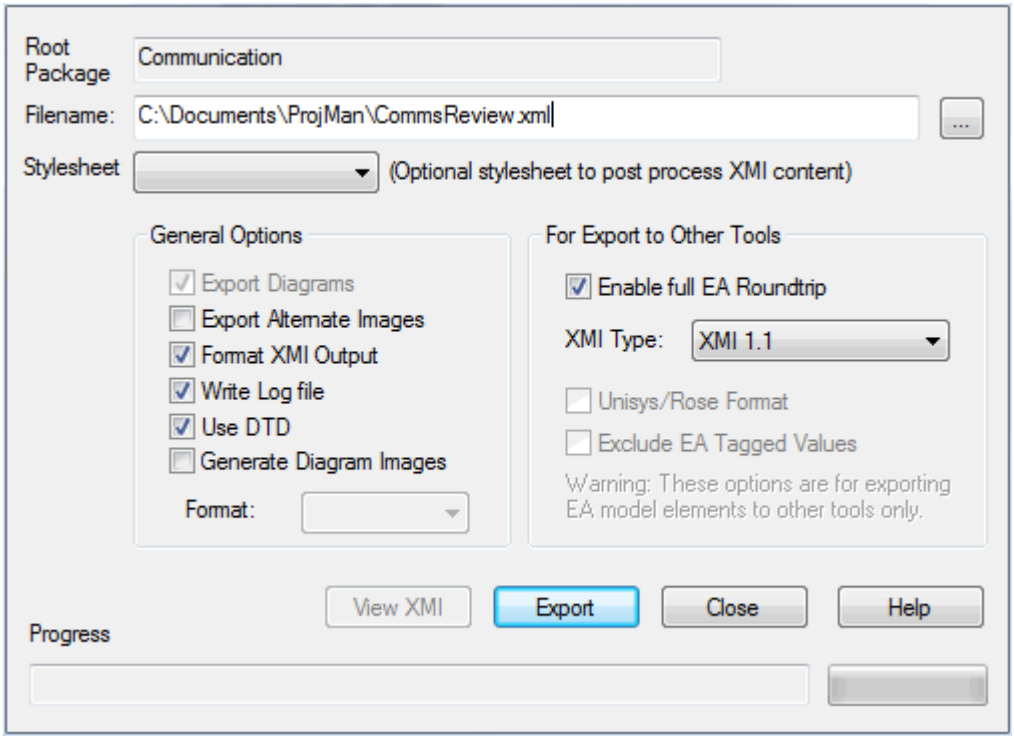
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Export XMI** 183 permission to export to XMI. See *User Security in UML Models.*
- You can post-process the XMI content before saving the package to file, using a style sheet to convert the output to HTML, XSL, code, or other versions of XMI. If you want to do this, you must import the required style sheet into the project through the Resources window. (See *Using Enterprise Architect - UML Modeling Tool.*)

Export a Package to XMI

To export a package to XMI, follow the steps below:

1. In the Project Browser window, select the package to export.
2. Either:
   - Right-click and select the **Import/Export | Export Package to XMI** context menu option, or
   - Select the **Project | Import/Export | Export Package to XMI** menu option.

   The Export Package to XMI dialog displays.



3. In the **Filename** field, type the directory path and filename into which to output the XMI file.
4. If required, in the **Stylesheet** field click on the drop-down arrow and select a stylesheet to post-process XMI content before saving to the file.
5. Select the **Export Diagrams** checkbox to export diagrams in the file.
6. Select the **Export Alternate Images** checkbox to export the alternative images used in the diagrams.
7. Select the **Format XMI Output** checkbox to format output into readable XML (this takes a few more seconds at the end of the run).
8. Select the **Write Log file** checkbox to write a log of export activity (recommended); the log file is saved to the directory into which you export the XMI file.
9. If using XMI 1.1, select the **Use DTD** checkbox to use the UML1.3 DTD (recommended). Setting this option validates the correctness of the model and checks that no syntactical errors have occurred. For more information regarding the use of DTDs, see the *UML DTD* 109 topic.
10. Leave the **Enable full EA Roundtrip** checkbox selected to keep data specific to Enterprise Architect in the XMI file.
11. In the **XMI Type**: field, click on the drop-down arrow and select the appropriate XMI format:
    - **XMI 1.1**, to generate output in XMI 1.1 format (necessary if you intend to use this file in a comparison with the model 210 at a later time - see *Baseline UML Models*)
    - **XMI 2.1**, to generate output in XMI 2.1 format.
10. Select the **Unisys/Rose Format** checkbox to export in Rose UML 1.3, XMI 1.1 format.
11. Select the **Exclude EA Tagged Values** checkbox to exclude Enterprise Architect-specific information from the export to other tools.
12. Click on the **Export** button.

Notes:

- XMI 2.1 exported by Enterprise Architect 7.0 (or later) might not be correctly imported into earlier versions of Enterprise Architect.
- 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.

Important:

When exporting and importing with XMI 1.0 with Enterprise Architect, some loss of data occurs due to the limitations of XMI 1.0.

8.2 Import from XMI

You can import a package from an XMI (XML based) file. This enables you to move Enterprise Architect Model elements between models, for distributed development, manual version control 116 and other benefits.

Important:

- When you import an XML file over an existing package, ALL information in the current package is deleted first. Before you import the XML file, please make sure you do not have important changes that you do not want to lose.
- If you are *importing* an 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.

Note:

In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Import XMI** 183 permission to import packages from XMI. See *User Security in UML Models.*

You can import the following formats:

- UML 1.3 (XMI 1.0)
- UML 1.3 (XMI 1.1)
- UML 1.4 (XMI 1.2)
- UML 2.0 (XMI 2.1)
- UML 2.1 (XMI 2.1)
- MOF 1.3 (XMI 1.1)
- MOF 1.4 (XMI 1.2)

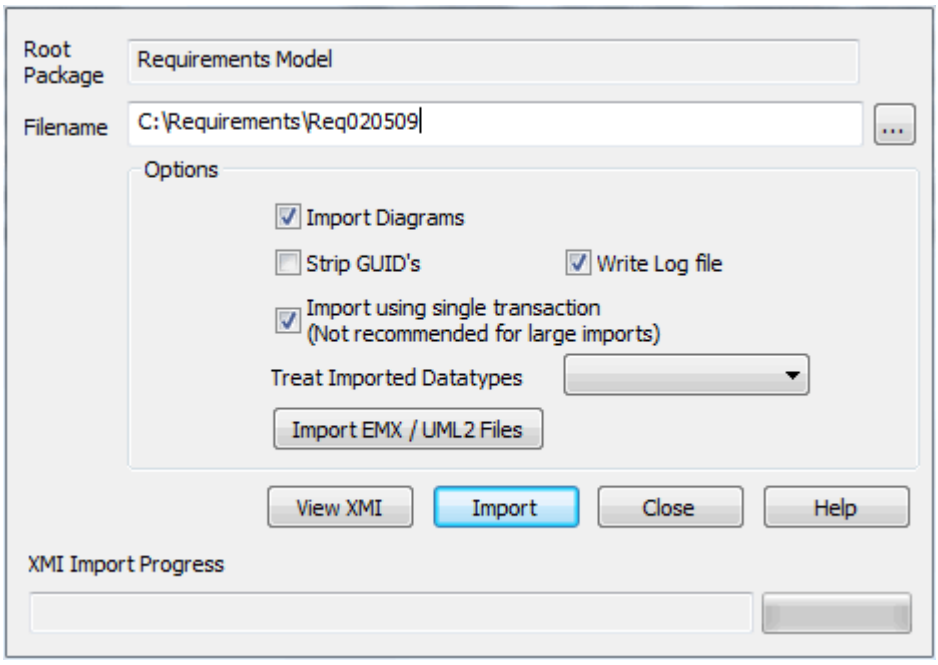
Enterprise Architect can also import the *.emx and *.uml2 files 108 generated by tools such as Rational Software Architect (RSA) and Rational Software Modeler (RSM).

Import From XMI

To import a package from XMI, follow the steps below:

1. In the Project Browser window, select the package into which to import the file.
2. Either:
   - Right-click and select the **Import/Export | Import Package from XMI** context menu option, or
   - Select the **Project | Import/Export | Import Package from XMI** menu option.

   The Import Package from XMI dialog displays.



Note:

To import .emx or .uml2 files, click on the **Import EMX / UML2 Files** button. Go to Import EMX/UML2 Files 108.

3. In the **Filename** field, type the directory path and filename from which to import the XMI file.
4. Select the **Import diagrams** checkbox to import diagrams.
5. Select the **Strip GUIDs** checkbox to remove Universal Identifier information from the file on import. This enables the import of a package twice into the same model; the second import requires new GUIDs to avoid element collisions.
6. Select the **Write log file** checkbox to write a log of import activity (recommended); the log file is saved in the directory from which the file is being imported.
7. **Import using single transaction** defaults to selected; if the import encounters locking issues, or if you are importing a large XMI file, deselect the checkbox to import the data items separately and identify

problem items without blocking the whole import.

8. If you are importing from Rose XMI 1.1, click on the **Treat Imported Datatypes** drop-down arrow and select the datatypes to add to the model.
9. Click on the **Import** button.

8.3 Import EMX/UML2 Files

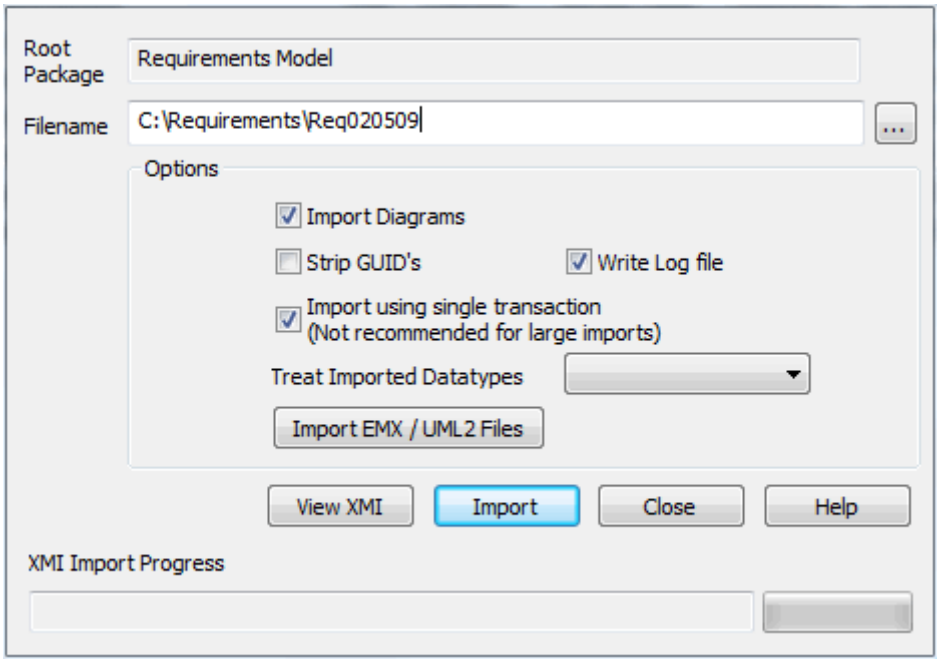
Rational Software Architect (RSA) enables you to add many UML models under a single root. These models can have cross references between them. However, RSA cannot save the entire root as one file; it saves each UML model as a separate EMX file. This means that an EMX file with cross-references is not self-contained as it references elements in another EMX file.

In releases earlier than release 7.0, Enterprise Architect treats each EMX file as a separate model and hence does not allow for cross-references between them. From release 7.0, Enterprise Architect enables these cross-references. You therefore have the option of importing a single EMX/UML2 file or a group of EMX/UML2 files. This option enables you to select a group of related files and import them together, thereby retaining the cross-references between the different files.

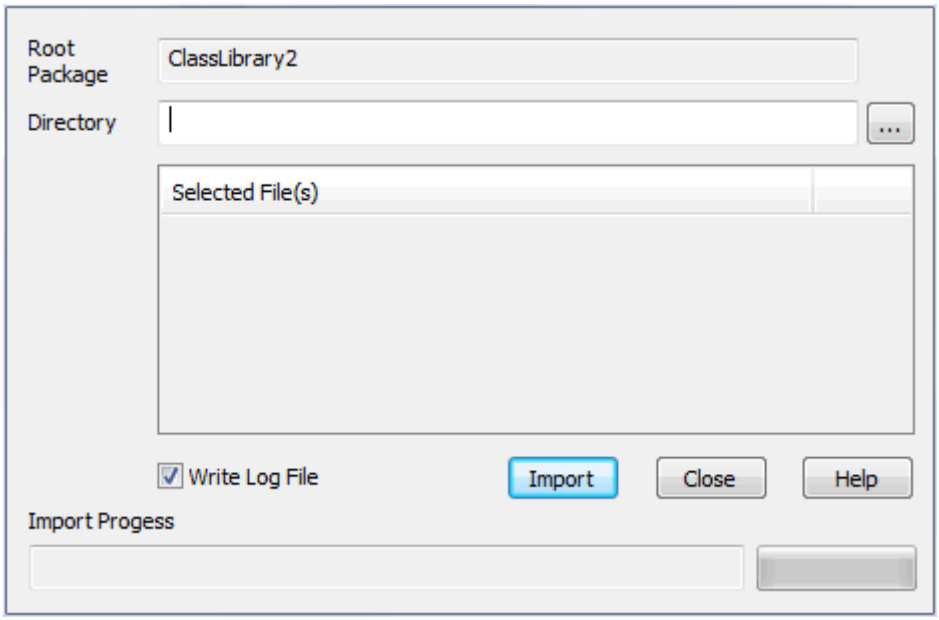
To import single or multiple *.emx /*.uml2 files into Enterprise Architect, follow the steps below:

1. In the Project Browser window, select the package into which to import the file.
2. Either:
   - Right-click and select the **Import/Export | Import Package from XMI** context menu option, or
   - Select the **Project | Import/Export | Import Package from XMI** menu option.

   The Import Package from XMI dialog displays.



3. Click on the **Import EMX / UML2 Files** button. The Import Package from XMI dialog redisplays, formatted for .EMX/.UML2 file imports.



4. Click on the **[...]** (Browse) button next to the **Directory** field. The Select Import EMX / UML2 File(s) dialog displays, which enables you to select multiple files.
5. Select the file or files (use **[Ctrl]+**click or **[Shift]+**click to select several files) and click on the **Open** button. The Import Package from XMI dialog redisplays; the Selected File(s) panel lists the selected files.
6. Select the **Write Log File** 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 name *import.log*.
7. Click on the **Import** button. Enterprise Architect indicates the progress of the import in the **Import Progress** field.

8.4 Limitations of XMI

Whilst XMI is a valuable means of specifying a UML model in a common format, it is relatively limited in the amount of additional information it can tolerate using the standard syntax. 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 2.0 and UML 2.1 format. Note that round-tripping model elements using XMI (for example, to version control or for controlled package) is only possible using XMI 1.1/UML 1.3 - Enterprise Architect format, which uses the additional Tagged Values to store the UML 2.0 information.

Notes for Exporting to Rose and Other Tools

- There are also discrepancies in the Unisys/Rose implementation with regard to spelling mistakes 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 like Rational Rose; for example, Note Links are not supported, and State Operations import but do not appear in diagrams. In addition, 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 Rose simple and conforming as closely as possible to the UML 1.3 specification.

8.5 The UML DTD

When you import or export Enterprise Architect packages to XMI, the import or export process can be validated using a Data Type Definition (DTD). The XML parser uses this document 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.

Several DTDs for XMI/UML exist. The OMG defines a standard UML1.3 DTD for use in XMI 1.1. Enterprise Architect uses an extension of this with some additional element extensions for non-standard UML types, such as testing details.

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. You must ensure the UML_EA.DTD file is in the current XML output path (generated by default).

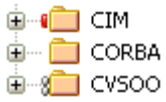
8.6 Controlled Packages

Controlled packages are a powerful means of 'externalizing' parts of an Enterprise Architect model. Using controlled packages you can:

- Support widely distributed development by having team members submit packages in the form of XML for import into a central Enterprise Architect repository.
- Support version control 130, by writing model elements in XML text files suitable for version control using standard version control software. Using XMI this way 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.
- Support 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.

Package XML is standard XMI-compliant output that can be loaded into any XML viewer, or used by any XML-based tool to perform manipulations and extracts, such as document or code generators.

Controlled packages appear in the Project Browser with a small colored rectangle to the left of the package icon, as shown below for the CIM package:



A controlled package is a package configured to save and load in XML format to a named file. The XML output is UML1.3 compliant XMI, with Enterprise Architect extensions to support diagrams and additional model elements.

Note:

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.

Important

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.

8.6.1 Controlled Package Menu

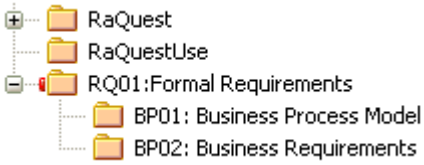
The **Package Control** sub-menu is available from the Project Browser **Package** context menu. This menu is for a package that itself is not under version control (but that might contain child packages that are under version control). For a package that is directly under version control, see the *Package Version Control Menu* 157 topic in *Version Control Within UML Models Using Enterprise Architect*.

Menu Option & Function Keys	Use to
Configure (various settings for the package) [Ctrl]+[Alt]+[P]	Display the Package Control Options 112 dialog, which enables you to specify whether this package (and its child packages) is controlled and which file it is controlled through.
Manage Baselines [Ctrl]+[Alt]+[B]	Create a Baseline 208 of the current package, or compare the current package with a previous Baseline. See *Baseline UML Models*.
Check In Branch	For the selected branch of the model, (that is, the selected package and all of its child packages) display the Select Packages to Check In 163 dialog, listing all version controlled packages *within* that branch that are checked out to you. You can then select packages in the displayed list, to be submitted for check-in. See *Version Control Within UML Models Using Enterprise Architect*.
Check Out Branch	For the selected package, check out the package and recursively check out all of its contained sub-packages 164. Retrieves the latest version of the packages from the central repository, overwriting the current packages. After check out, the packages are available for editing. See *Version Control Within UML Models Using Enterprise Architect*.
Save package to file [Ctrl]+[Alt]+[S]	Save a controlled package 114 to an XMI file.
Load package from file [Ctrl]+[Alt]+[L]	Load 114 a previously-saved XMI file.
View package XMI [Ctrl]+[Alt]+[X]	Display the package XMI after the package has been exported to XMI.
Compare with XMI file	(Package not under version control.) Compare 210 the current package with a previously-saved XMI file of the package. See *Baseline UML Models.*
Compare with Controlled Version	(Version controlled package.) Compare 210 the current package with the head revision of the version-controlled package file. See *Baseline UML Models.*
Add Branch to Version Control	Apply version control to all packages within a selected model branch 165, in a single operation. See *Version Control Within UML Models Using Enterprise Architect.*
Export as Model Branch	Export 166 a newly created model branch from your own private copy of a model. See *Version Control Within UML Models Using Enterprise Architect.*
Import a Model Branch	Retrieve 166 a model branch and import it into either the source model or another model. See *Version Control Within UML Models Using Enterprise Architect.*
Get package (for version control)	Enables you to gain access from packages in the version-controlled 130 repository that is currently available in your model. See *Version Control Within UML Models Using Enterprise Architect.*
Get All Latest (for version control)	Retrieve the latest version 130 of the package from the repository. Available only for packages that are checked in. See *Version Control Within UML Models Using Enterprise Architect.* The alternative option **Get Latest** - if displayed - is not intended for sharing .EAP files and should only be used when users have their own individual databases.
Re-synch Status With VC Provider	(Version controlled package.) Update the version control status

Menu Option & Function Keys	Use to
	value recorded for the selected package in the Enterprise Architect project to match the value reported by the version control provider 169, without performing an XMI import or export.
Version Control Settings	Display the Version Control Settings 135 dialog. See *Version Control Within UML Models Using Enterprise Architect.*
Update Package Status	Provide a bulk update on the status of a package. This includes status options such as **Proposed**, **Validate** and **Mandatory**. See *Project Management with Enterprise Architect*.

8.6.2 Configure Packages

Before you can use a controlled package, you must configure it with options such as the filename to save to/ load from, the type of export and the version number. Once a package is configured and marked as controlled, it is displayed in the Project Browser with a small colored rectangle next to the package icon, indicating it is a controlled package. In the example below, the *RQ01: Formal Requirements* package is a controlled package.



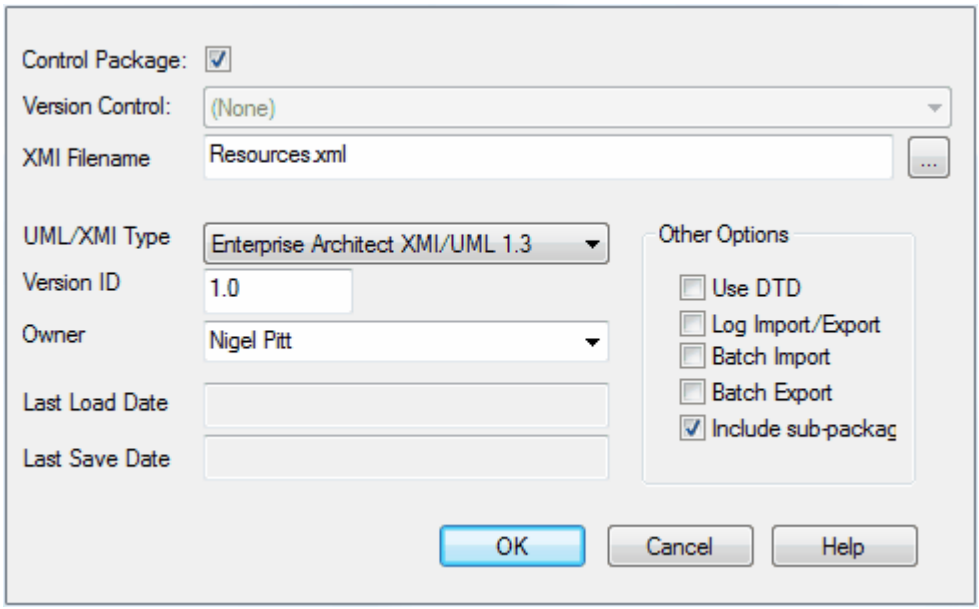
Note:

In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Configure Packages** 183 permission to configure controlled packages and package properties. See *User Security in UML Models.*

Configure a Controlled Package

To configure a controlled package, follow the steps below:

1. In the Project Browser, right-click on the package to control or configure. The context menu displays.
2. Click on the **Package Control | Configure** menu option. The Package Control Options dialog displays.



3. Set the required options, as follows:
   - Select the **Control Package** checkbox to indicate that this is a controlled package
   - Click on the **Version Control** drop-down arrow and select the version control repositories; this connects the package to a specific version control configuration
   - In the **XMI Filename** field, type or browse for the path and XMI file for importing and exporting XMI files. The field accepts Local Path Substitution strings; for example, use an XMI local path definition where:

     myLocalPath="C:\Documents and Settings\John\Desktop\EA Models"

     Then %myLocalPath%\CIM.xml is equivalent to C:\Documents and Settings\John\Desktop\EA Models\CIM.xml
   - In the **UML/XMI Type** field, click on the drop-down arrow and select the type of XMI generated; options include Enterprise Architect XMI/UML 1.3, Rational Rose/Unisys UML 1.3 and Generic XMI 1.0/UML 1.3 - currently only Enterprise Architect UML 1.3 is supported for complete import/export round tripping of packages
   - In the **Version ID** field, type the version ID number
   - In the **Owner** field, type or select the name of the package owner
   - If required, click on the **Use DTD** checkbox to use a Data Type Definition (DTD)
   - If required, click on the **Log Import/Export** checkbox to log import and export activity to a log file
   - If required, click on the **Batch Import** checkbox to mark the package as a Batch Import package
   - If required, click on the **Batch Export** checkbox to mark the package as a Batch Export package
   - If required, click on the **Include sub-package** checkbox to *deselect* it, to include only the immediate contents of the package in an XMI export (XMI stubs); this is available only for an XMI 1.1/UML 1.3 export. If you leave the checkbox selected, the entire sub-package hierarchy of this branch is included in the export.
4. Click on the **OK** button to set the Package Control options.

Note:

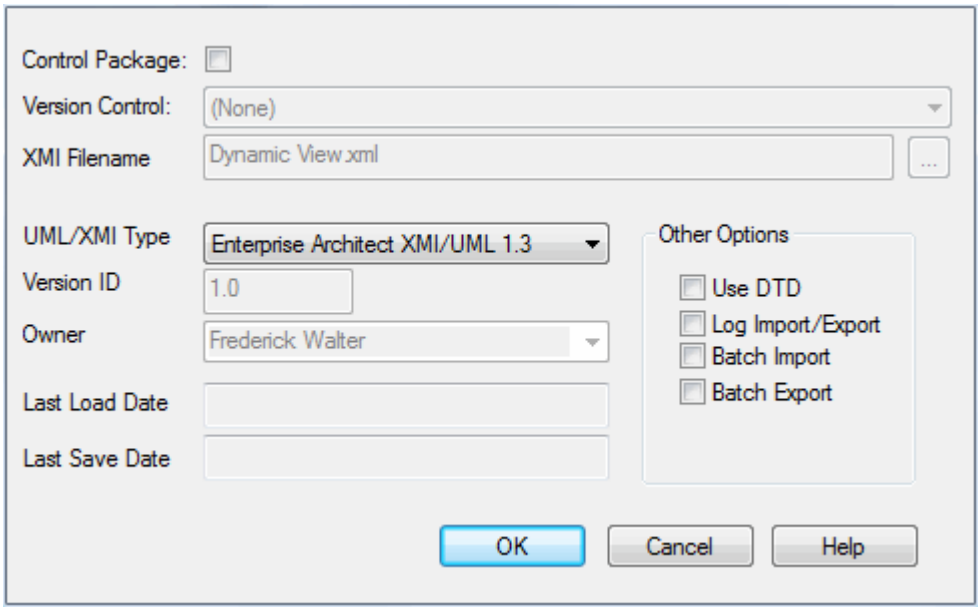
For batch import, the file date of the XMI file is stored. You can bypass the batch import if the file date of the last import matches that of the current file (that is, there is no change).

8.6.3 Remove Package from Control

If required, you can remove the control from a package. Before removing the package control, you must check in 162 the package if it is being used for version control. See *Version Control Within UML Models Using Enterprise Architect*.

To remove control from a package:

1. In the Project Browser window, right-click on the controlled package. The context menu displays.
2. Select the **Package Control | Configure** menu option, or press **[Ctrl]**+**[Alt]**+**[P]**. The Package Control Options dialog displays.
3. Deselect the **Control Package** checkbox. If the package is under version control, this sets the **Version Control** field to **(None)**.



4. Click on the **OK** button to remove package control.

Package control for the selected package has now been removed.

Note:

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.

8.6.4 Save a Package

You can save a controlled package to an XMI file. Once you have correctly configured 112 the package, follow the steps below:

1. In the Project Browser window, right-click on the package to save. The context menu displays.
2. Select the **Package Control | Save Package to File** menu option.
3. The export process executes automatically according to your configured preferences, overwriting any existing file.

Note:

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.

8.6.5 Load a Package

Using the *Controlled Packages* facility, you can save and load packages to a named file. If a package has been marked for control it is displayed in the Project Browser with a red rectangle to the left of the package icon. If you have previously saved a controlled package, you can reload it using the **Load package from file**

menu option.

To load a controlled package, follow the steps below:

1. In the Project Browser window, right-click on the package to load. The context menu displays.
2. Select the **Package Control | Load package from file** menu option.
3. If you have configured the package control details, Enterprise Architect prompts you to confirm the import.

> **Warning:**
>
> 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.

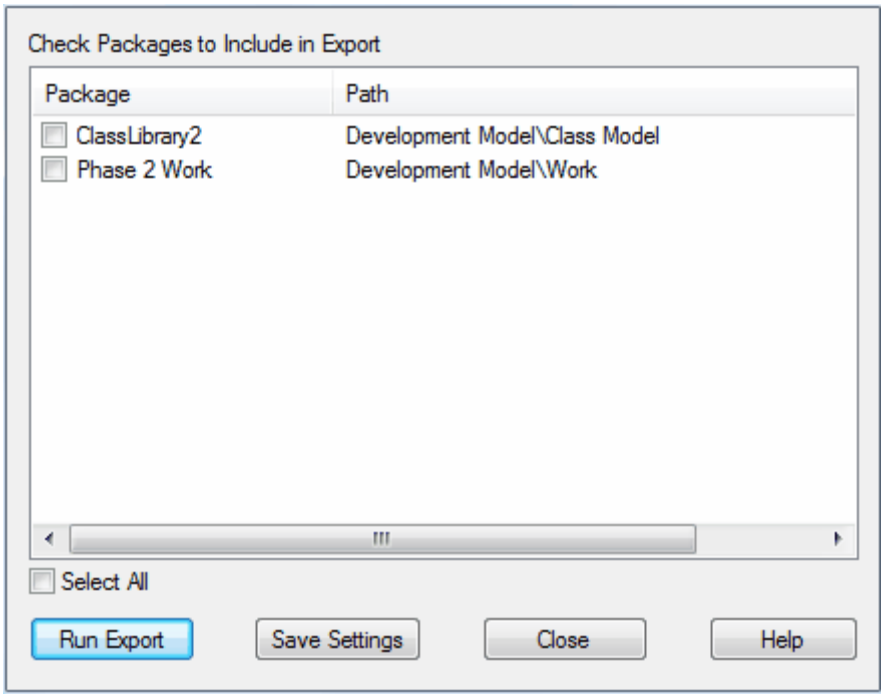
4. Click on the **Yes** button to confirm the import. The current package is deleted and the saved package is imported.

8.6.6 Batch XMI Export

You can export a group of controlled packages in one step, using the *Batch XMI Export* facility.

To export a group of controlled packages, follow the steps below:

1. Select the **Project | Import/Export | Batch XMI Export** menu option. The Batch XMI Export dialog displays.



2. Select the checkbox against each package to include in this export run. Select the **Select All** checkbox to select all packages in the list.
3. To save this configuration as the default, click on the **Save Settings** button.
4. Click on the **Run Export** button.

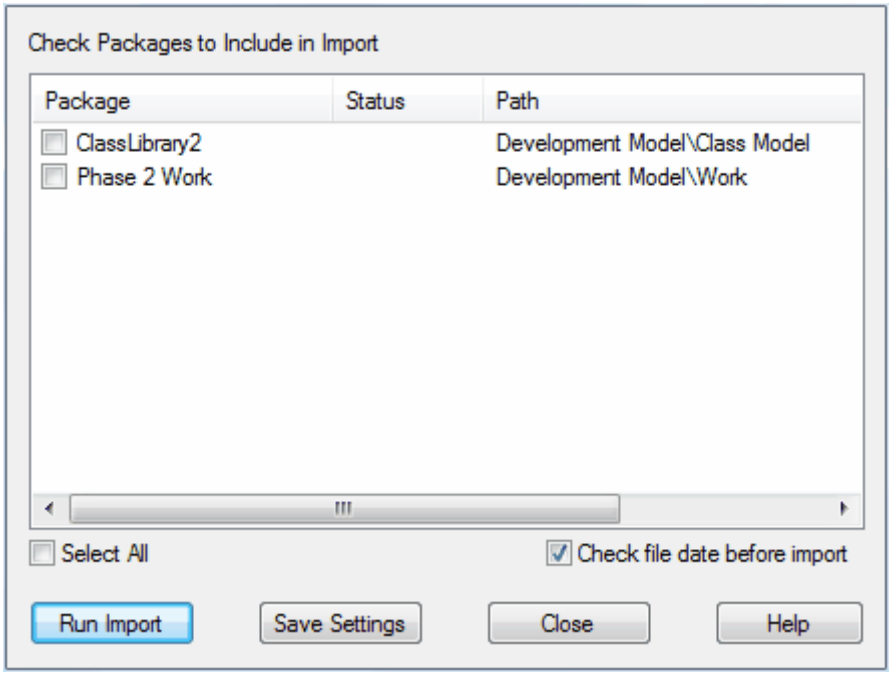
Enterprise Architect cycles through each checked package and exports it using the options specified in the Controlled Package 112 dialog. As long as a valid filename exists, Enterprise Architect exports the package to XML and proceeds to the next package.

8.6.7 Batch XMI Import

You can import a group of controlled packages in one step, using the Batch XMI Import facility.

To import a group of controlled packages, follow the steps below:

1. Select the **Project | Import/Export | Batch XMI Import** menu option. The Batch XMI Import dialog displays.



2. Select the checkbox against each package to include in the import. Select the **Select All** checkbox to select all packages in the list.

Tip:

To avoid re-importing the same module multiple times, select the **Check file date before import** checkbox. Enterprise Architect then does not import a file if the last import file date matches that of the one currently on disk.

3. To save this configuration as the default, click on the **Save Settings** button.
4. Click on the **Run Import** button. Enterprise Architect cycles through the packages and imports each selected package.

As Enterprise Architect processes each package, it updates the **Status** column against each package name on the Batch XMI Import dialog.

- If the import is successful, Enterprise Architect updates the package status to **Imported**.
- If the import is unsuccessful, Enterprise Architect updates the package status to **Not Imported**.

Common reasons for an import to fail include:

- The package not being correctly configured
- The last import file date matches the import date of the file currently on disk.

8.6.8 Manual Version Control with XMI

You can use XMI to support version control by writing model elements in XML text files suitable for use with standard version control software. Using XMI in this manner enables you to manually connect to third-party version control software outside the Enterprise Architect environment. Enterprise Architect internally supports the configuration of version control through SCC and CVS configurations.

To use XMI for version control, you must first:

1. Select suitable packages in the Project 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.

When Versioning is Required

Continue working on a package until versioning is required then follow the steps below:

1. Check out the package XMI file from the version control system.
2. Save the relevant package using the controlled package support.
3. Check the package back into the version control system.

Recover an Earlier Version

To recover an earlier version, follow the steps below:

1. Save the current version first (important, because the package is completely deleted during the import process) and manually update the version control system if necessary.
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.

9 CSV Import and Export

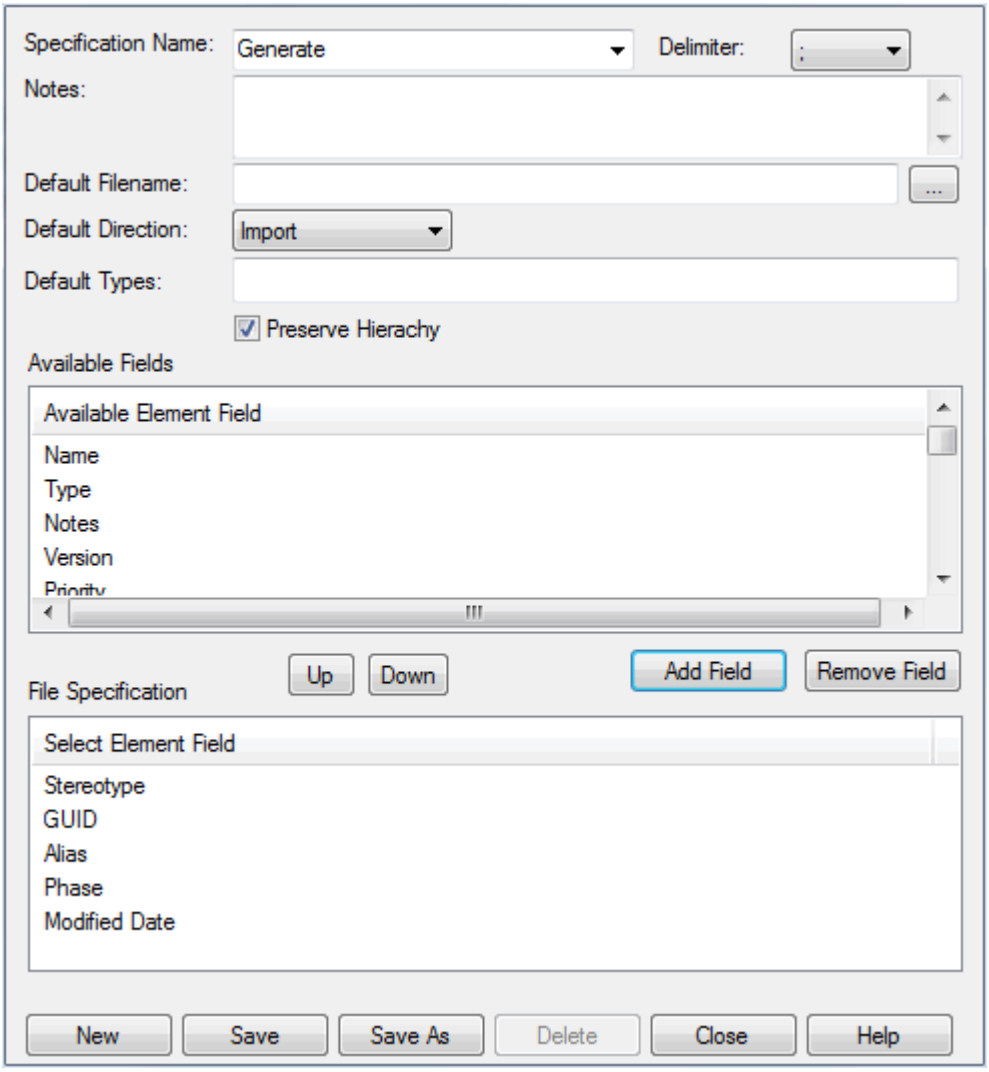
You can import [123] and export [121] information about Enterprise Architect elements in CSV format. You must define CSV specifications [118] to do this, because the specification defines what types of value from the spreadsheet are to be imported, and how the information is translated between the spreadsheet and Enterprise Architect.

9.1 CSV Specifications

To import [123] and export [121] element data from Enterprise Architect using CSV files, you must first set up one or more file specifications. A file specification lists the fields from the spreadsheet in the order they are imported or exported, the filename (optional) and the delimiter between columns. Once you have defined one or more specifications, one can be selected in the CSV Import/Export dialog as the current specification to apply during an import or export action.

CSV only imports and exports elements (within packages) and their properties; items such as Class attributes cannot be imported or exported through this mechanism. XMI [104] provides a solution to this limitation, as does use of the Automation Interface. See the *Enterprise Architect Object Model* topic in *SDK for Enterprise Architect.*

To define a specification, select the **Project | Import/Export | CSV Import/Export Specifications** menu option. The CSV Import/Export Specification dialog displays.



The CSV Import/Export File Specification dialog provides the following functionality:

Option	Use to
Specification Name	Select the unique name for this specification.
Delimiter	Specify the character delimiter to use between record fields. **Note:** If a field contains an instance of the delimiter, the field is exported wrapped in " (quotation marks) and all instances of " in the field are doubled (that is, " becomes "").
Notes	Record a brief description of the specification.
Default Filename	Select the default filename.
Default Direction	Set the default action - **Import** or **Export**. A specification can be used in either direction, but this enables you to set the default type.
Default Types	Limit the element types being exported, by entering a comma-separated list: for example, *class,requirement,component,node,object*.

Option	Use to
	Note: If you specify element types, ONLY elements of those types are exported or imported. Therefore, in order to enable the **Preserve Hierarchy** option to operate (if selected) you must include Package as an element type. Otherwise there are no packages in which to preserve the hierarchy. If you do not specify any default element types, all elements including Packages are exported or imported and the hierarchy can be preserved.
Preserve Hierarchy	Include fields generated by Enterprise Architect to embed/reconstruct the package hierarchy. See the *Using Preserve Hierarchy* 120 section for more details.
Available Fields	Select from a list of possible record fields, not yet allocated.
File Specification	List the record fields (in the order they are plotted across the spreadsheet) already assigned.
Add Field	Move all selected fields in the top list to the bottom list.
Remove Field	Move all selected fields in the bottom list back to the available list.
New	Create a new specification.
Save	Save changes to the currently selected specification.
Save As	Save the current specification with a new name.
Delete	Delete the current specification.
Close	Close this dialog.

Note:

In **Available Fields** and **File Specification**, the record fields **Created Date** and **Modified Date** are not set when imported from CSV.

Using Preserve Hierarchy

When selected, the **Preserve Hierarchy** option inserts two fields into the CSV specification that are:

- automatically populated by Enterprise Architect on export and
- used to reconstruct the exported package's hierarchy upon import.

Field	Description
CSV_KEY	A unique identifier for the current element. **Note:** 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.

However, 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

Note:

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

9.2 CSV Export

It is possible to export information about Enterprise Architect elements in CSV format. Once you have defined a CSV export specification 118 it is possible to write out major element attributes to a CSV text file.

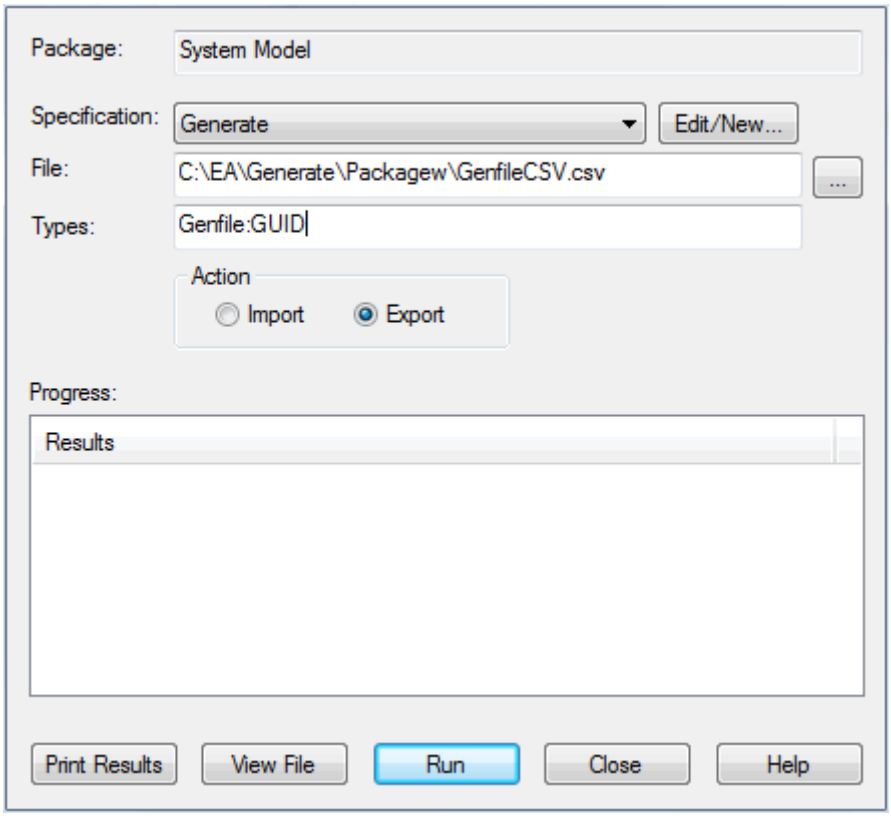
Note:

In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have *both* **Export XMI** 183 and **Import XMI** 183 permissions to use the **CSV Import/Export** option. See *User Security in UML Models.*

Export Data in CSV Format

To export data in CSV format, follow the steps below:

1. In the Project Browser, right-click on the package containing the elements to export.
2. Select the **Import/Export | CSV Import/Export** context menu option. The CSV Import/Export dialog displays.



3. Set the required options, as outlined below:

Option	Use to
Package	Confirm the name of the current selected package.
Specification	Specify the name of the export specification 118 to use.
Edit/New	Edit the export specification or create a new one.
File	Specify the filename to export to.
Types	List the element types to export: leave blank for all, or enter a comma-separated list of types. **Note:** If you specify element types, ONLY elements of those types are exported. Therefore, in order 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.
Action	Select the **Export** radio button to export to file.
Print Results	Print out the result list.
View File	View the resultant CSV file with the default Windows application for CSV files.
Run	Perform the export.
Close	Exit this dialog.

9.3 CSV Import

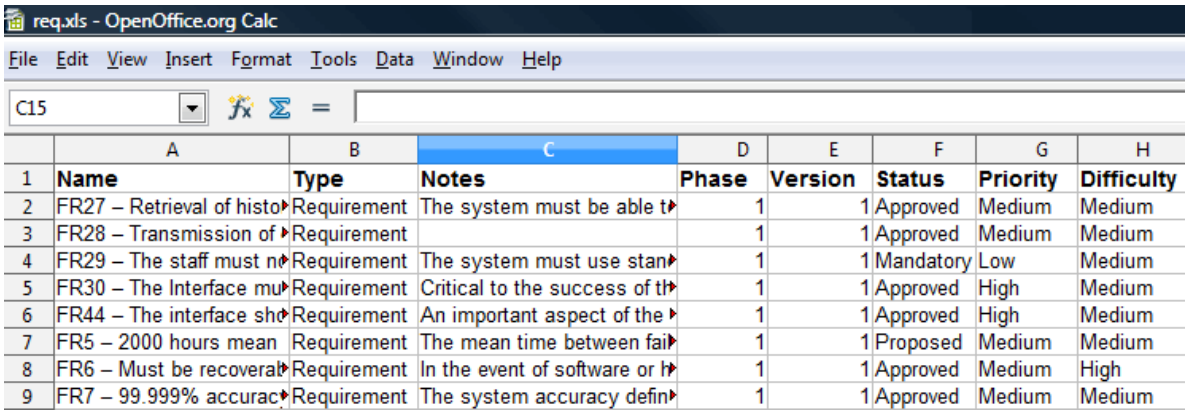
It is possible to import information about Enterprise Architect elements in CSV format. Once you have defined a CSV import specification 118 you can read in major element attributes from a CSV text file.

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have *both* **Export XMI** 183 and **Import XMI** 183 permissions to use the **CSV Import/Export** option. See *User Security in UML Models.*
- 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.

When importing, Enterprise Architect checks the specification to see if there is an **EAGUID** field included. If there is, Enterprise Architect attempts to locate the element identified by the EAGUID and, if successful, updates the current element rather than creating a new one. If no **EAGUID** field is defined, or Enterprise Architect cannot locate the identified element, a new element is created and placed in the current package. Note that 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*. (See the *Enterprise Architect Object Model* section in *SDK For Enterprise Architect.*)

The format and content of the source data file should resemble the following:


req.xls - OpenOffice.org Calc

File Edit View Insert Format Tools Data Window Help

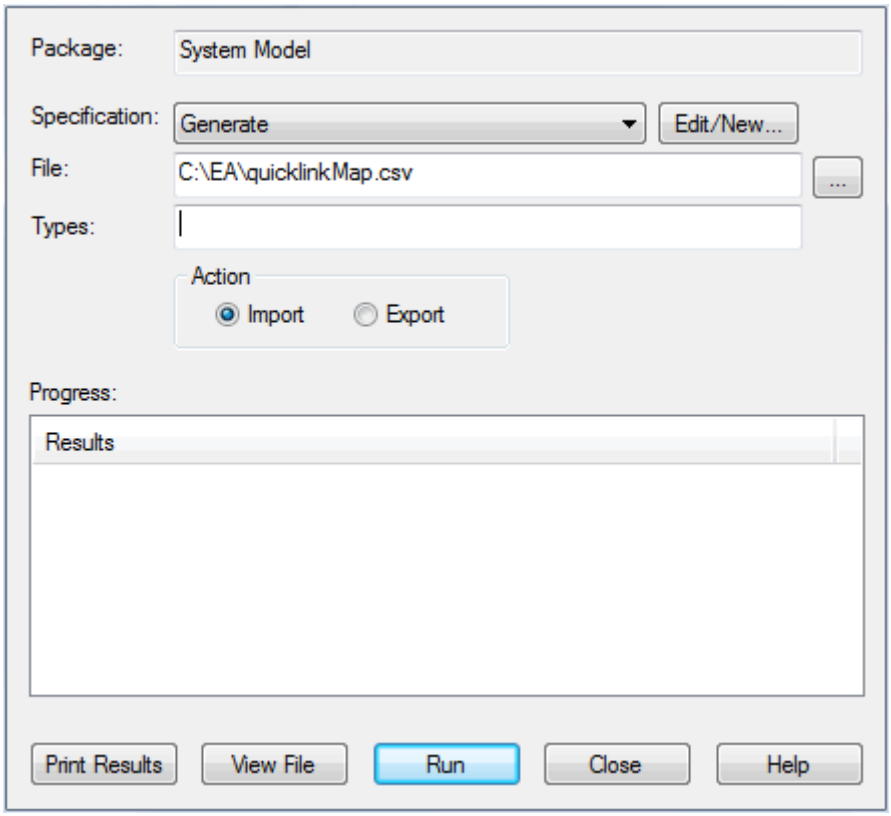
C15

	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 th	1	1	Approved	High	Medium
6	FR44 – The interface sh	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 recoverab	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

Import Data in CSV Format

To import data in CSV format, follow the steps below:

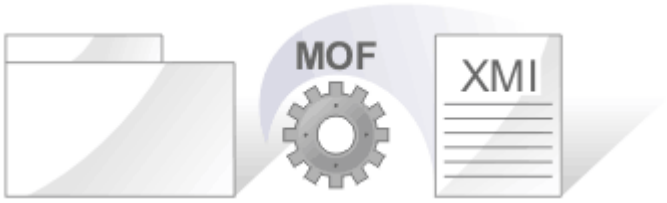
1. In the Project Browser, right-click on the package to import into.
2. Select the **Import/Export | CSV Import/Export** context menu option. The **CSV Import/Export** dialog displays.



3. Set the required options; as outlined below:

Option	Use to
Package	Confirm the name of the current selected package.
Specification	Specify the name of the import specification 118 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.
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	Exit this dialog.

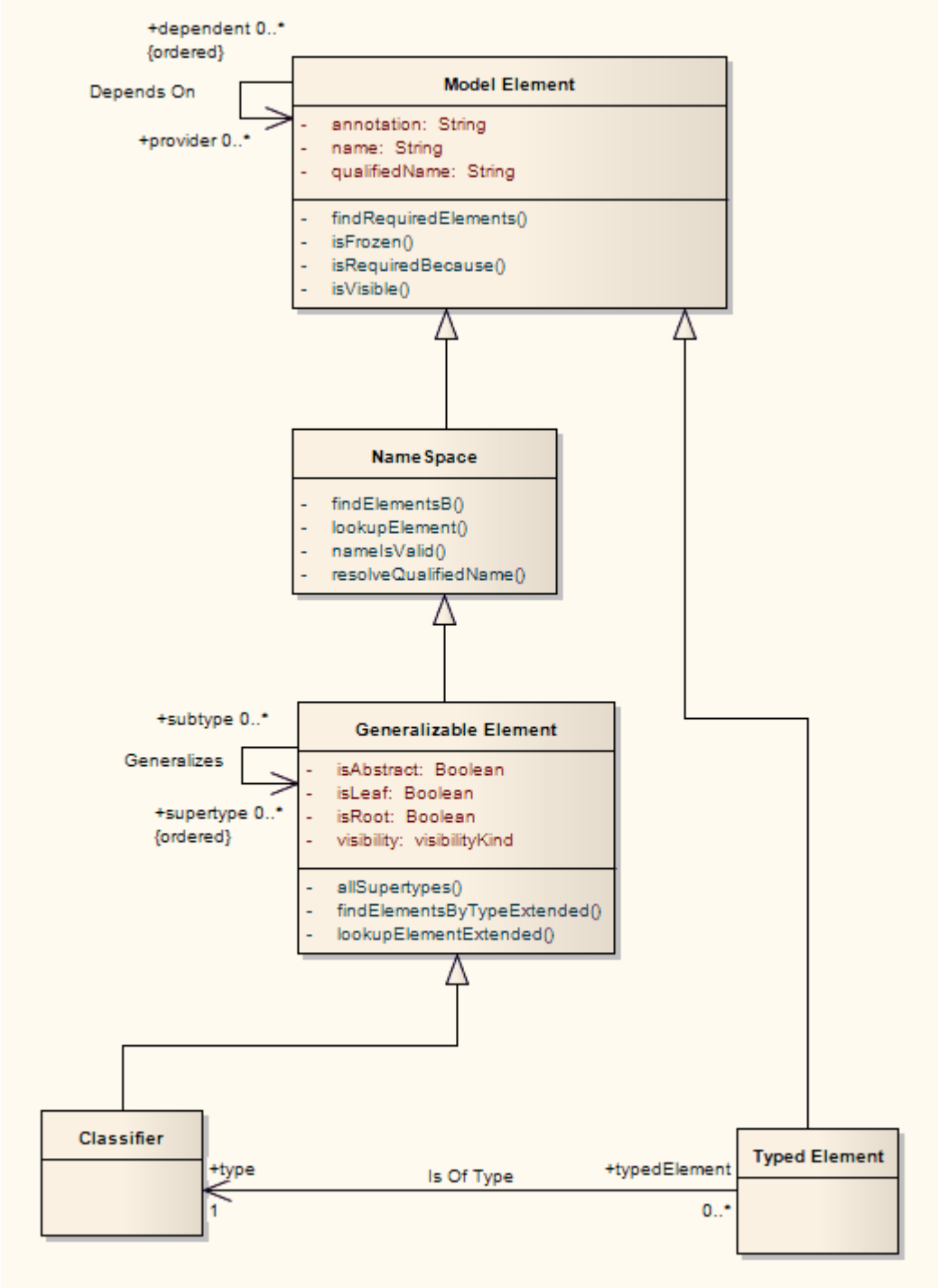
10 MOF



Enterprise Architect offers support for exporting packages to XMI under the *Meta-Object Facility (MOF)* 1.3 and 1.4 standards. MOF models are created by assigning the stereotype *metamodel* to the package. MOF models can be exported to MOF 1.3 or MOF 1.4 XMI file specification.

Background Knowledge

MOF is an Object Management Group (OMG) standard that originated in the UML, when the OMG required a Meta-Modeling architecture to define the UML. MOF is designed as a four-layered architecture, as illustrated in the following diagram.



Because of the similarities between the MOF-model and UML structure models, MOF meta-models are usually modeled as UML Class diagrams (see *The UML Dictionary)*. You can also use the Metamodel page of the Enterprise Architect UML Toolbox to create MOF model elements and connectors (see *Using Enterprise Architect - UML Modeling Tool*). A supporting standard of MOF is XMI, which defines an XML-based exchange format.

MOF is a closed, strict meta-modeling architecture; every model element on every layer is strictly an instance of a model element of the layer above. MOF only provides a means to define the structure or abstract syntax of a languages or of data.

Simplified, MOF uses the notion of Classes, as known from object orientation, to define concepts (model elements) on a meta-layer. These Classes (concepts) can then be instantiated through objects (instances) of

the model layer below. Because an element on the M2 layer is an object (instance of an M3 model element) as well as a Class (an M2 layer concept) the notion of a *clabject* is used. *Clabject* is a merge of the words *Class* and *object*.

Another related standard is OCL, which describes a formal language that can be used to define model constraints by means of predicate logic.

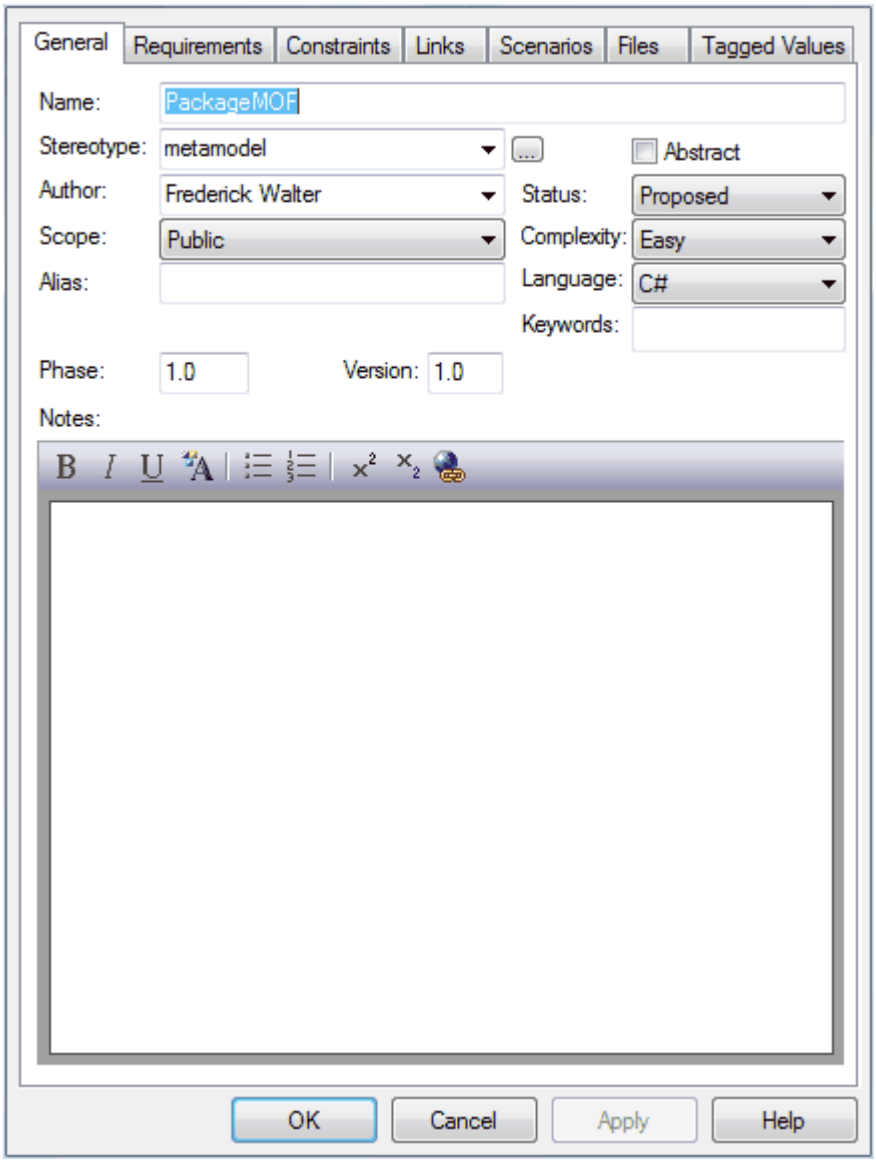
See Also

- Getting Started 127
- Export MOF to XMI 128

10.1 Getting Started

MOF diagrams are Class diagrams that are contained in packages with a metamodel stereotype. To create a MOF diagram, follow the steps below.

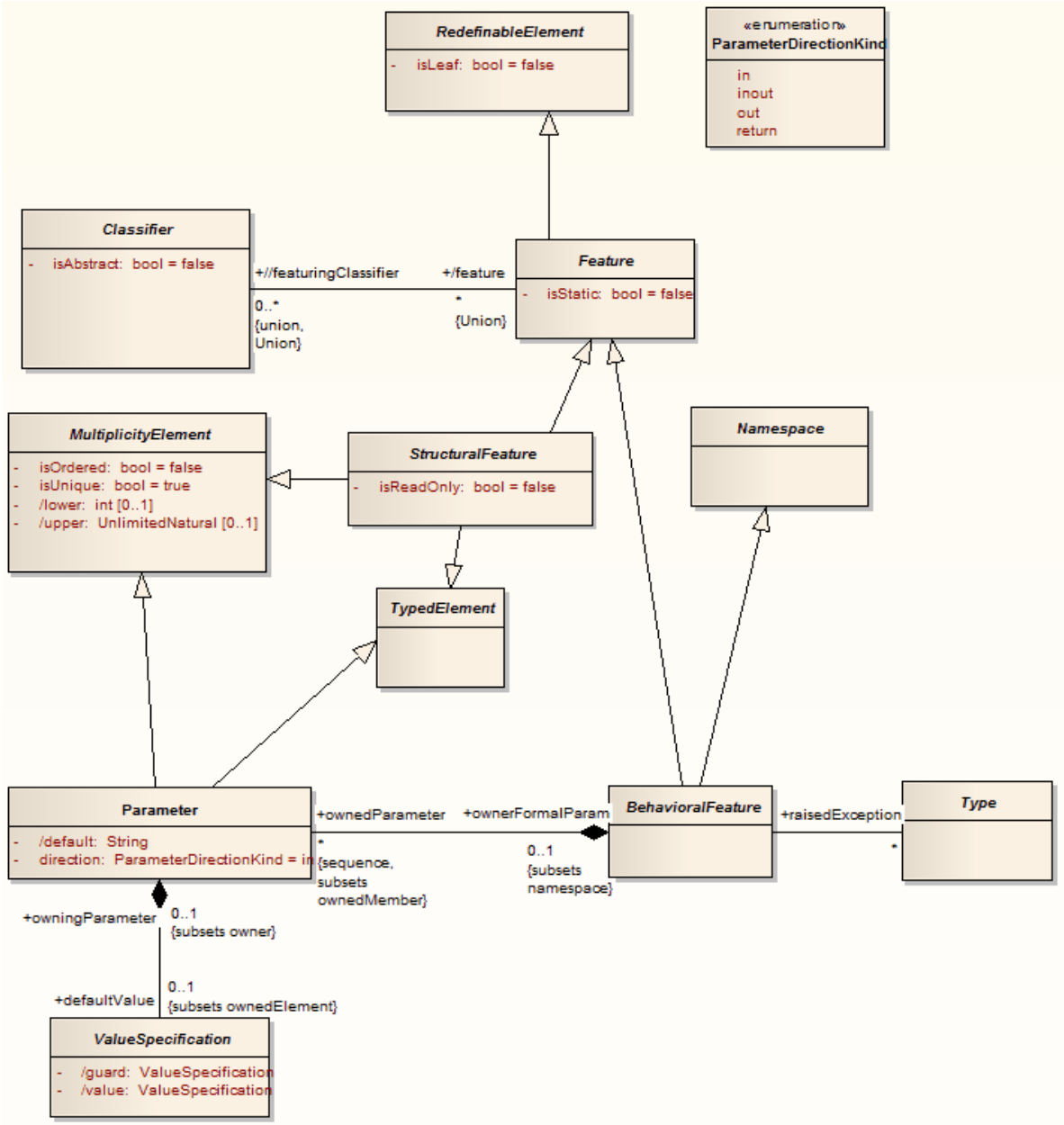
1. Create a package to contain your MOF elements.
2. Double-click on the package name to display the Properties dialog.



3. In the **Stereotype** field type the value **metamodel**.
4. Click on the **OK** button.

5. Right-click on the package in the Project Browser and select the **Add | Add Diagram** context menu option. Create a Class diagram (the default diagram).
6. Give your MOF Class diagram an appropriate name.
7. In the Enterprise Architect UML Toolbox, select the **More tools | Metamodel** menu option and add the required Metamodel elements to the diagram.

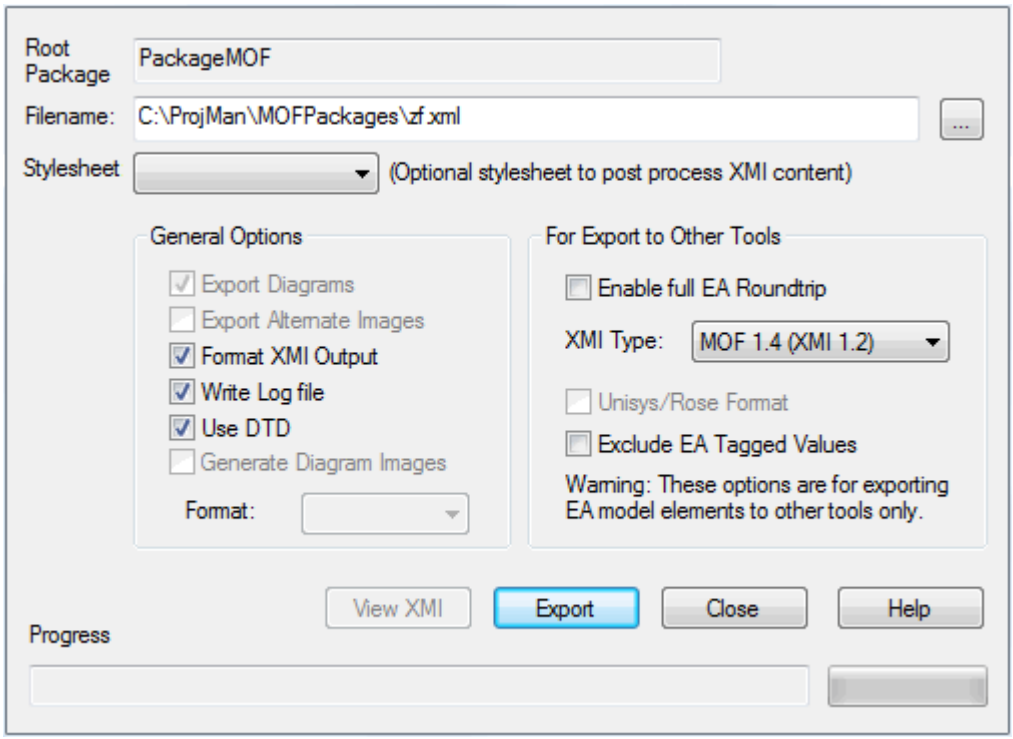
The following is an example of a MOF diagram. A MOF diagram can typically contain Package, Class, Enumeration and Primitive elements, and Generalization, Association, Compose and Aggregate relationships.



10.2 Export MOF to XMI

Once you have created your MOF diagram you can export the diagram to XMI, specifying the MOF 1.3 or MOF 1.4 standard.

1. Right-click on the package in the Project Browser. The context menu displays.
2. Select the **Import/Export | Export Package to XMI file** menu option. The Export Package to XMI dialog displays:



3. In the **Filename** field, type a name for the XMI file.
4. De-select the **Enable full EA Roundtrip** checkbox.
5. In the **XMI Type** field, click on the drop-down arrow and select **MOF 1.3** or **MOF 1.4.**
6. Click on the **Export** button and wait until the **Progress** bar reads **100%**.
7. Once your file has been created, you can view it by clicking on the **View XMI** button.

MOF diagrams exported to XMI can be imported using the regular import XMI features of Enterprise Architect. See Import from XMI 106.

11 Version Control

Enterprise Architect supports version control of packages and their component sub-packages to a central version control repository. You can place any individual packages, View nodes or model root nodes under version control.

Features

Version Control provides two key facilities:

- Coordinating sharing of packages between users
- Saving a history of changes to Enterprise Architect packages, including the ability to retrieve previous versions.

System Requirements

Version controlled packages are packages that have been configured for use with version control software. To use version control in Enterprise Architect, a third-party source-code control application is required that controls access to and stores revisions of the controlled packages. Enterprise Architect supports the following version control applications:

- Subversion, which is available from http://www.subversion.tigris.org/
- CVS, which is available from http://www.march-hare.com/cvspro/
- Microsoft Team Foundation Server
- SCC-compatible products; all version control products that provide a client that complies with the Microsoft Common Source Code Control standard, version 1.1 or higher.

  The following products are SCC-compatible and are known to successfully integrate with Enterprise Architect:

- Accurev	Tested by Sparx	
- Borland Star Teams		Users report success
- ClearCase		Users report success
- MS Visual Source Safe	Tested by Sparx	
- MS TFS-SCC	Tested by Sparx	
- MKS Source Integrity	Tested by Sparx	
- Perforce	Tested by Sparx	
- Serena Dimensions		Users report success
- Serena Change Manager		Users report success
- Snapshot CM	Tested by Sparx	
- SourceGear Vault	Tested by Sparx	
- Source Offsite	Tested by Sparx	

  Products that do not appear in the list should still integrate successfully with Enterprise Architect, if there is a client available for that product that complies with the MS SCC API specification.

Set-Up

Before using Enterprise Architect's version control facility, your version control software must be installed on each machine on which it is intended to be used.

Typically there are:

- A server component that manages a version control repository, and
- Client components on the workstations that Enterprise Architect uses to communicate with the server.

A version control client must be installed on every machine where you run Enterprise Architect and want to access your version control system. Once the version control software has been installed and configured, you must define a Version Control Configuration within Enterprise Architect, to use your installed version control product.

Note:

Sparx Systems strongly urge you not to manipulate version controlled package files outside of Enterprise Architect. It is possible to leave the package files in a state that Enterprise Architect cannot recognize.

Usage

There are four basic ways in which the version control facility might be used:

Use	Description
Single Shared model	Users share an Enterprise Architect model, stored in a central .EAP file or DBMS repository. This configuration enables you to view changes to other users' packages without explicitly having to check them out, but by simply refreshing your view of the model. • Version control regulates access to packages, and maintains package revision history.
Multiple Private models	An Enterprise Architect model is created by a single user who configures it for version control. The model file is then distributed to other users, with each user storing their own private copy of the model. • Users update their model's packages through version control • Version control regulates access to packages, and maintains package revision history • Other users' new packages are retrieved using the **Get Package** menu option.
Shared packages	Individual users create separate Enterprise Architect models but share one or more packages. • Users share packages through version control.
Standard packages	A company might have a standard set of packages which are broadly shared (on a read-only basis). • Individual users retrieve packages with the **Get Package** menu option.

For a discussion of how each of these arrangements might be used, see the Version Control white paper: http://www.sparxsystems.com/resources/whitepapers/index.html.

Version Control Indicators

Packages under version control are identified in the Project Browser by icons that indicate the current status of the package.

Icon	Indicates that...
	This package is controlled 110 and is represented by an XMI file on disk. Version control either is not being used or is not available. You can edit the package.
	This package is version controlled and checked out 162 to you, therefore you can edit the package.
	This package is version controlled and not checked out to you, therefore you cannot edit the package (unless you check the package out).

Icon	Indicates that...
	This package is version controlled, but you checked it out whilst not connected to the version control 169 server. You can edit the package but there could be version conflicts when you check the package in again.

For example, below, *CVSOO* and *CVSPackage* are configured for version control. *CVSOO* is checked out to you, and *CVSPackage* is not.

See Also

- Version Control Basics 132
- Apply Version Control To Models 132
- Version Control and Team Deployment 133
- Version Control Menu 134
- Version Control Setup 134
- Use Version Control 157
- Offline Version Control 169

11.1 Version Control Basics

The Lock-Modify-Unlock Solution

Many version control systems use a lock-modify-unlock model to address the problem of different authors in a shared source overwriting each other's work. In this model, the version control repository allows only one person to change a file at a time, and access is managed using locks. Harry must lock a file before he can begin making changes to it. If Harry has locked a file, Sally cannot also lock it, and therefore cannot make any changes to that file. All she can do is read the file, and wait for Harry to finish his changes and release the lock. After Harry unlocks the file, Sally can take her turn in locking and editing the file.

The Copy-Modify-Merge Solution

Subversion, CVS and a number of other version control systems use a copy-modify-merge model as an alternative to locking. In this model, each user's client contacts the project repository and creates a personal working copy—a local reflection of the repository's files and directories. Users then work simultaneously and independently, modifying their private copies. In due course, the private copies are merged together into a new, final version. The version control system often assists with the merging, but ultimately a person is responsible for making it happen correctly.

When Locking is Necessary

While the lock-modify-unlock model is generally considered a hindrance to collaboration, there are still times when locking is necessary.

The copy-modify-merge model is based on the assumption that files are contextually merge-able: that is, the files in the repository are line-based text files (such as program source code). But for files with binary formats, such as artwork or sound, it is often impossible to merge conflicting changes. In these situations, it really is necessary for users to take strict turns in changing the file. Without serialized access, somebody ends up wasting time on changes that are ultimately discarded.

11.2 Apply Version Control To Models

All Enterprise Architect models are stored in databases - even the .EAP file is a database. In simple, version control terms, the model is a single entity of binary data. It is not practical to apply version control to the database as a whole. Being binary data, it would require the use of the lock-modify-unlock model 132 of version control, which would mean that only a single user at a time could work on any given (version controlled) model.

To overcome this limitation, Enterprise Architect exports discreet units of the model - the packages - as XMI package files, and it is these XMI files, not the .EAP file, that are placed under version control. The XMI file format used by Enterprise Architect dictates that they too be treated as binary files (therefore it is not possible to merge the XMI files either); however, by splitting the model into much smaller parts, this approach enables many users to work on separate parts of the model simultaneously.

When a user checks-out 162 a package, Enterprise Architect sends a command to the version control system to check-out the equivalent XMI file. The version control system then puts the latest revision of the file into the user's working copy directory, overwriting any previous revision of the file in that directory. Enterprise Architect then imports the package file into the model, updating the contents of the existing package in the model.

When checking-in 162, Enterprise Architect exports the package as an XMI file, overwriting the existing local working copy of the file. The new file is then checked-in to the version control system.

Nested Version Controlled Packages

Nested version controlled packages result in much smaller XMI files being exported for parent packages, as the parent packages' XMI files do not contain any content for the version controlled child packages.

Version Control of nested packages 137 together with a model structure having small individual packages also provides greater scope for multiple users to work concurrently, as individual users are locking much smaller parts of the model.

Notes:

- **Do not place your .EAP files under version control**, as this creates problems for you.
- Most version control systems mark their controlled files as read only, unless they are specifically checked-out to you.
- The .EAP file is an MS Jet database, and Enterprise Architect **must** be able to open this file for read/write access when you load your model. (Enterprise Architect displays an error message and fails to load the model if it is read-only.)

11.3 Version Control & Team Deployment

Team deployment and the use of version control is discussed in two Sparx Systems white papers, available on the Sparx Systems web site:

- http://www.sparxsystems.com/WhitePapers/Version_Control.pdf
- http://www.sparxsystems.com/downloads/whitepapers/EA_Deployment.pdf

A brief summary of the process is provided below:

1. Install your version control product.
2. Create a version control repository.
3. Create a version control project to be used with your Enterprise Architect project, and check-out a working copy of the project into a local folder. (You must do this for every team member that is accessing the version controlled packages, whether you are using a single shared model or each team member stores his own private copy of the model.)
4. Within Enterprise Architect, define a version control configuration 134 to provide access to the working copy files. Again, each user must do this on their own workstation, as the details are stored within the Windows registry.
5. Configure packages 160 within the Enterprise Architect model for version control. That is, apply version control to individual packages.
6. Check-out and check-in packages 162 as required.

Note:

The name of the version control configuration must be the same across all machines. That is, all version control access to a given Enterprise Architect package must be through version control configurations with the same name, across all models and all users. (It is possible to use multiple version control configurations within the same model, so different packages can still use different version control configurations within the same model, as long as any given package is always accessed via the same version control configuration.)

The easiest way to perform step 4, (throughout the team), is to have one user set up version control on the

model and then share that model with the rest of the team.

- In Shared Model deployment, all users connect to a single instance of the model database, so the model is shared automatically.
- In Private Model deployment, it is easiest to distribute copies of the original model (after version control has been set up) to all other members of the team.

Whenever you open a model (Private or Shared 131) that uses a version control configuration that is not yet defined on your workstation, Enterprise Architect prompts you to complete the definition for that configuration. This typically means specifying the local working copy directory and maybe choosing the version control project associated with this Enterprise Architect project.

Once this has been done, the version controlled packages that already exist in the model are ready for use.

Version Control Branching

Currently, Enterprise Architect does not support Version Control Branching. Work-arounds to achieve similar results might be possible for certain version-control products; contact Sparx Support for advice:

- Registered users - http://www.sparxsystems.com/registered/reg_support.html
- Trial users - support@sparxsystems.com.

11.4 Version Control Menu

You access the **Version Control** menu through the **Project | Version Control** menu option. It provides the following options:

Menu Option & Function Keys	Use to
Configure Current Package [Ctrl]+[Alt]+[P]	Display the Package Control Options 160 dialog, which enables you to specify whether this package (and its children) is version controlled, and which version control configuration applies.
Version Control Settings	Display the Version Control Settings dialog 135.
Validate Package Configurations	Test the validity 161 of the version control settings associated with each version controlled package within your current model.
Re-Synch Statuses of All Packages	Resynchronize the version control status of packages 169 as recorded in your Enterprise Architect project when they are out of synchronization with the version control status reported by your version control provider. The function acts on all version controlled packages within the Enterprise Architect project, updating the values recorded in the project to match the values reported by the version control provider, without performing any XMI import or export.
Work Offline	Work independently of the version control server 169, if it is unavailable to you.

11.5 Version Control Setup

Before using Enterprise Architect's version control facility, your version control product must be installed on each machine where it is intended to be used. Version Control products supported by Enterprise Architect include MS Team Foundation Server, Subversion, CVS or any other version control product that provides an MS SCC-compliant interface.

- Subversion is available from http://www.subversion.tigris.org/
- CVS is available from http://www.wincvs.org/.

Note:

If you are using the Corporate, Business and Software Engineering, System Engineering or Ultimate editions of Enterprise Architect with security enabled, you must also set up permissions to configure and use version control. See the *List of Available Permissions* 184 topic for further information.

Typically there should be:

- A server component that manages a version control repository
- Client components on the workstations that Enterprise Architect uses to communicate with the server.

A version control client must be installed on every machine where you run Enterprise Architect and want to access your version control system. Once the version control software has been installed and configured, to use your installed version control product you must define a Version Control Configuration within Enterprise Architect.

Version control can be assigned to individual packages, view nodes or root nodes in Enterprise Architect. Each package can only be linked to one Version Control Configuration at a time, although it is possible to connect multiple control configurations for each model. You can use the Version Control Settings dialog to set up a connection to your version control application.

To set the Version Control Configuration, select the **Project | Version Control | Version Control Settings** menu option, and see the *Version Control Settings Dialog* 135 topic.

See Also

- Version Control with SCC 137
- Version Control with CVS 141
- Version Control with Subversion 148
- Version Control with TFS 154

11.5.1 Version Control Settings Dialog

The Version Control Settings dialog enables you to specify the information required to create a Version Control Configuration, which can then be used to establish a connection to a version control provider. Enterprise Architect supports version control through MS Team Foundation Server, Subversion, CVS or any SCC-compliant version control product.

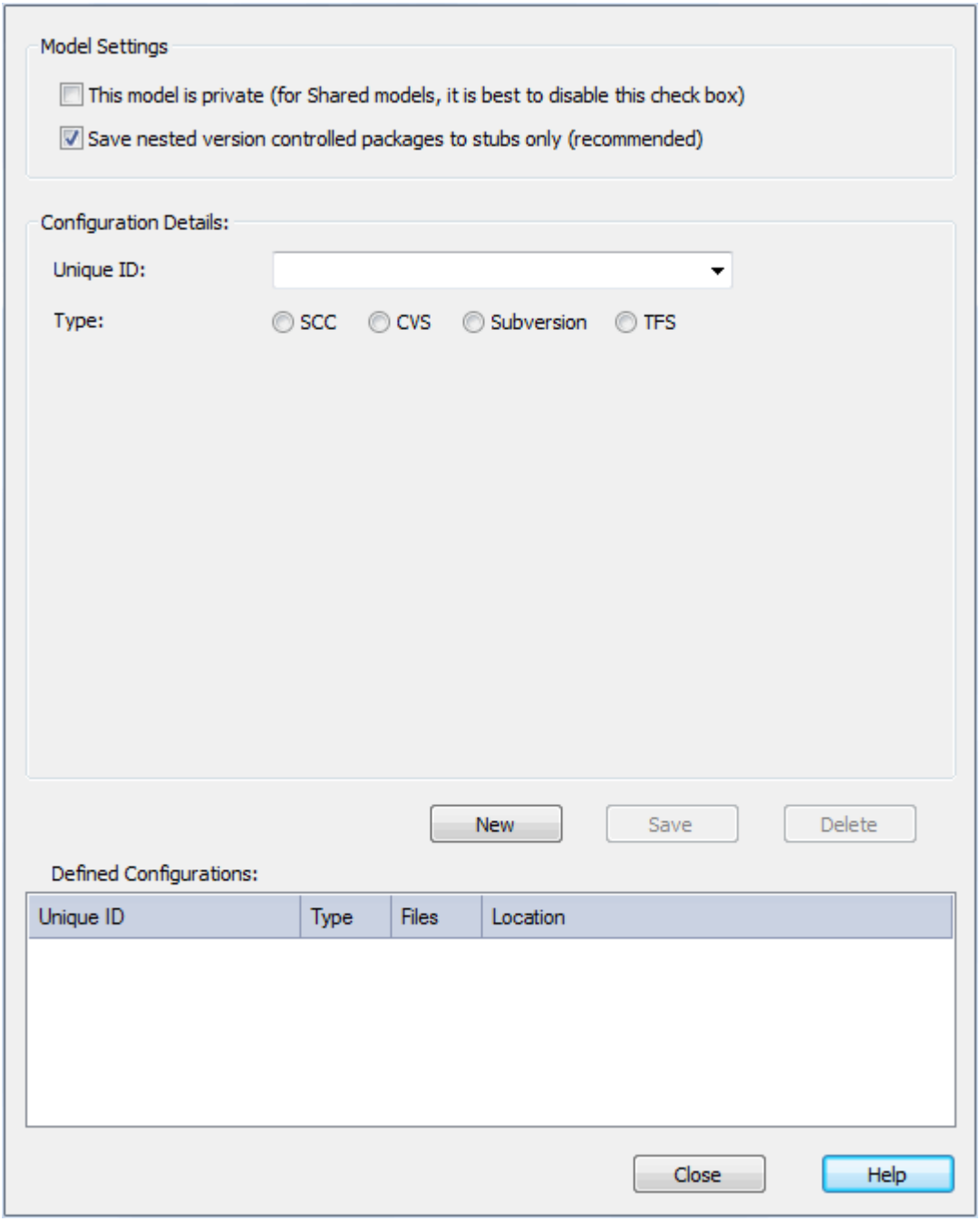
It is possible to use multiple version control configurations in the same Enterprise Architect model. It is also possible to use the same version control configuration across different models, to facilitate sharing 'standard' packages between those models, through the version control system.

Note:

In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Configure Version Control** 183 permission to set up version control options for the current model.

Setting Up Version Control

When you display the Version Control Settings dialog for the first time in any given model, it appears as shown below:



To begin defining a new version control configuration, follow the steps below:

1. Click on the **New** button.
2. In the **Unique ID** field, type a suitable name.
3. Against the **Type** field, click on the radio button for the version control product to connect to.

   At this point, the middle section of the dialog changes to display a collection of fields relating to the type of Version Control Configuration you are defining. Go to the relevant topic below:

   - Version Control with SCC 137
   - Version Control with CVS 141
   - Version Control with Subversion 148
   - Version Control with TFS 154

To import a previously defined configuration for use in the current model, follow the steps below:

1. Click on the **New** button.
2. In the **Unique ID** field, click on the drop-down arrow and select one of the previously defined version control configurations.
3. Click on the **Save** button to save the selected version control configuration in this model.

See Also

- Version Control Nested Packages 137

11.5.1.1 Version Control Nested Packages

In releases of Enterprise Architect later than version 4.5, when you save a package to the version control system only stub information is exported for any nested packages. This ensures that information in a nested package is not inadvertently over-written by a top level package.

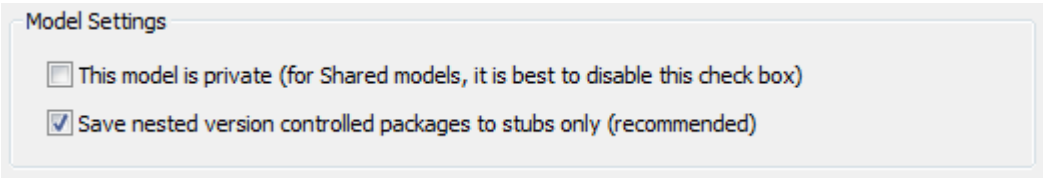
When checking out a package, Enterprise Architect does not modify or delete nested packages; only the top level package is modified.

As a consequence of this behavior, if you check out or get a version controlled package with nested packages not already in your model, you see stubs in the model for the nested packages only. If you select the **Get All Latest** option from the version control menu, Enterprise Architect populates these new stubs from the version control system.

Using the above technique you can populate a large and complex model from only the root packages, using **Get All Latest** to recursively iterate through the attached and nested packages.

This is a powerful and efficient means of managing your project and simplifies handling very large models, even in a distributed environment.

It is recommended you do not mix versions of Enterprise Architect later than version 4.5 with earlier versions when sharing a version controlled model. If this is necessary it is best to go to the Version Control Settings 135 dialog and deselect the **Save nested version controlled packages to stubs only** checkbox, setting Enterprise Architect to the pre-version 4.5 behavior (for the current model only).



11.5.2 Version Control with SCC

To set up an SCC version control configuration, you must:

- Set up the source code control provider with SCC, and
- Connect the Enterprise Architect model to version control with SCC.

See also, the topic on version control with SCC when *Upgrading at Enterprise Architect 4.5* 140.

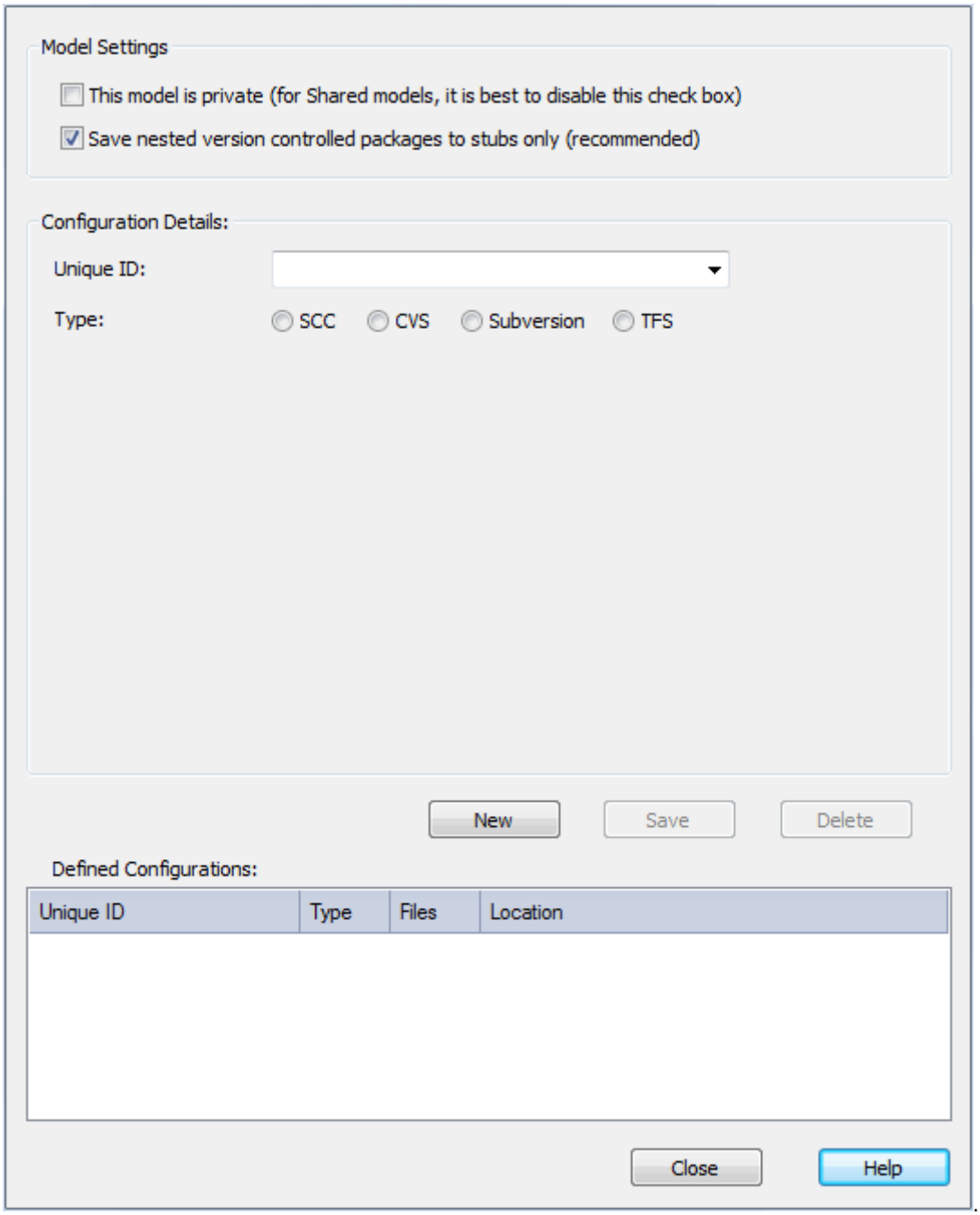
Set Up the Source Code Control Provider with SCC

To set up the third-party source code control provider, see the documentation provided with that application. A repository must be set up using the SCC provider and access to that repository must be available to all intended users.

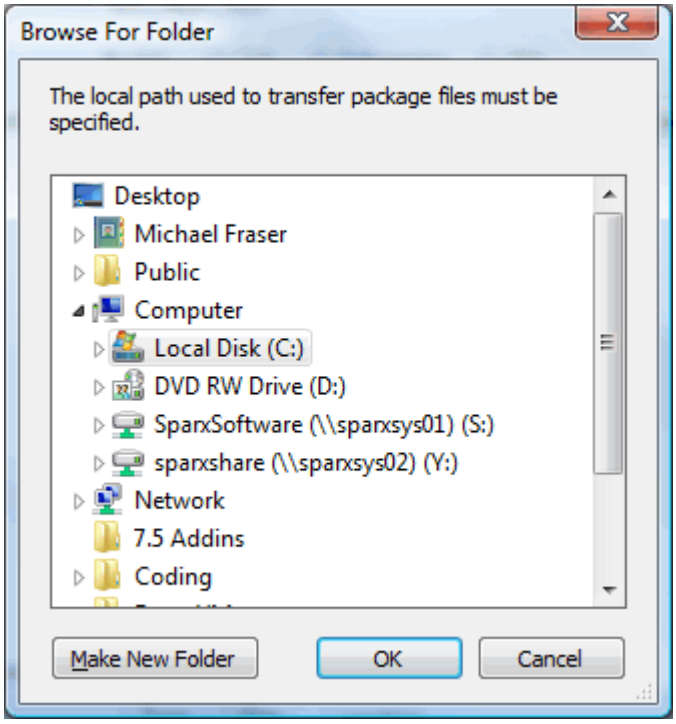
Connect an Enterprise Architect Model to Version Control with SCC

To connect an Enterprise Architect model to version control, follow the steps below:

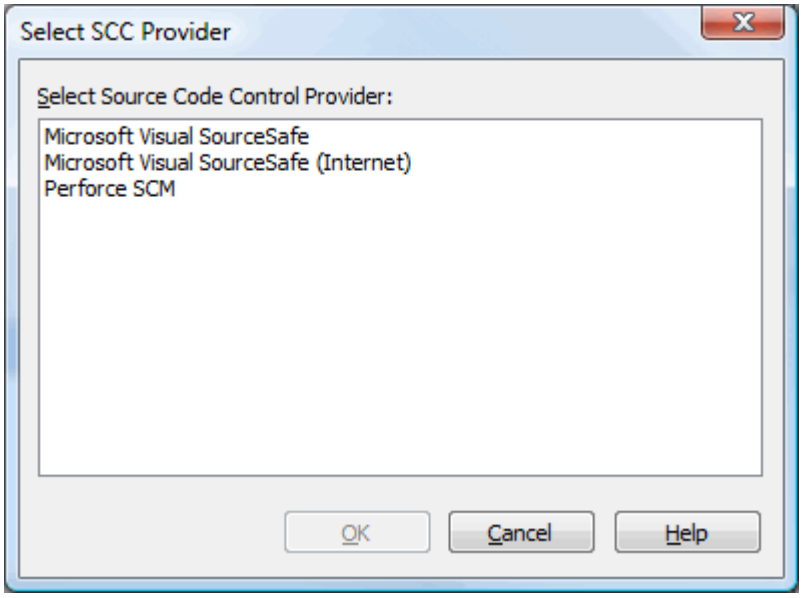
1. Open or create the Enterprise Architect model to place under version control.
2. Select the **Project | Version Control | Version Control Settings** menu option. The Version Control Settings dialog displays.



3. Click on the **New** button.
4. In the **Unique ID** field, type a suitable name. Click on the **SCC** radio button.
5. To the right of the **Local Project** path field, click on the **Select Path...** button. The Browse for Folder dialog displays.



6. Locate and click on the local folder in which to keep local working copies of the XML files to be stored in the Version Control repository.
7. Click on the **OK** button. The Select SCC Provider dialog displays.



Note:

All users of the shared database must specify the same SCC provider.

8. Click on an SCC provider, and click on the **OK** button to return to the Version Control Settings dialog.
9. Click on the **Save** button to save the configuration you have defined.

   The SCC provider is likely to prompt you for various details including the name of the project to connect to, and perhaps the user name to use when you log in.
10. The new configuration is added to the list in the Defined Configurations panel.

Note:

A new entry is also created in the Local Paths list, with the same ID as the new version control configuration. The **Local Path** entry records the Local Project path, for use in subsequent path substitutions.

11. When you have finished defining your version control configurations, click on the **Close** button.

For further information on the fields on the Version Control Settings dialog, see the following table.

Field	Use to
This model is private	Specify whether all users connect to a single shared copy of the model (for example, a DBMS) or each user connects to their own private copy of the model. When unselected (for shared models), the option disables the **File History - Retrieve** functionality when the selected package is checked out by another user. This prevents modifications that might have been made by the other user from being discarded through importing a prior revision from version control.
Save nested version controlled packages to stubs only	Set nested version controlled packages to stubs or fully expanded trees. Defaults to selected. For a full explanation of this option, see Version Control Nested Packages 137.
Unique ID	Specify a configuration name that readily distinguishes this configuration from other configurations. The unique ID is displayed as a selection in the list of Version Control configurations a package can connect to. You can also click on the drop-down arrow and select a previous version control configuration, providing the configuration is not in the current model.
Local Project Path	Specify the folder in which the XML files representing the packages are stored. This folder should already exist before it is specified here. Every PC using version control should have its own local SCC project folder, and this should not be a shared network folder. Particularly bear this in mind if you are creating a .EAP file that is to be shared (such as a SQL database).
Current User	Read only. Shows your user name as the user currently logged into the SCC provider.
SCC Provider	Read only. Shows the name of the provider specified in the database.
SCC Project	Read only. Shows the project selected during the initial setup of the connection to the SCC provider.

Note:

Sparx Systems strongly urge you not to manipulate version controlled package files outside of Enterprise Architect. It is possible to leave the package files in a state that Enterprise Architect cannot recognize.

11.5.2.1 Upgrade at Enterprise Architect 4.5

When a version-controlled project created under a release of Enterprise Architect earlier than 4.5 is opened in Enterprise Architect release 4.5 or later, you must identify the SCC connection with a new unique ID. You can assign a name to the existing SCC configuration or associate the project with a configuration that has previously been assigned a unique ID.

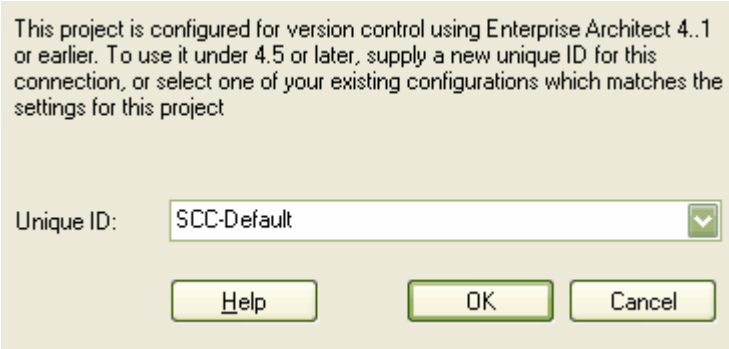
By having a unique ID for Version Control Configurations, you can assign a configuration quickly and efficiently using configurations that have been created previously for other version controlled repositories. This enables you to configure the many packages to use an existing version control repository; this can apply to packages created for more than just one model enabling a great deal of flexibility.

To upgrade an existing SCC version control project created before release 4.5, in Enterprise Architect release 4.5 or later, follow the steps below.

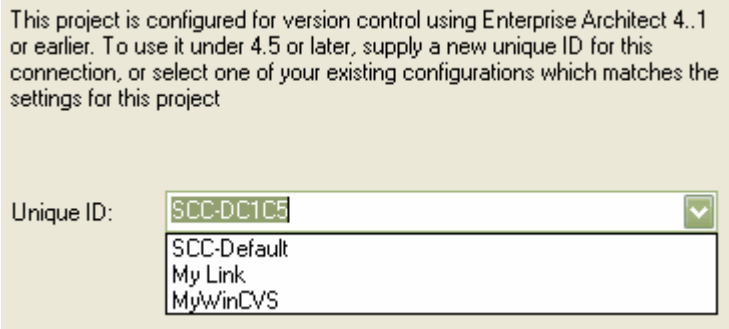
1. Open the project that has an SCC Version Control Configuration created in Enterprise Architect earlier

than version 4.5.

2. The Select or Create Unique ID for Version Control dialog prompts you to create an ID for an existing configuration or to choose a previously created one from the **Unique ID** drop-down list.
3. The existing SCC configuration is the initial value, represented by **SCC-XXXXX**; this number is not especially meaningful, therefore it is recommended that the configuration be given a meaningful name.



4. You can associate the version controlled package with a previously-defined configuration by selecting an existing configuration from the **Unique ID** drop-down list (if one exists).



5. After you have assigned the unique ID, click on the **OK** button to load the model.

11.5.3 Version Control with CVS

CVS is used to manage files and directories and is an open source, version control system. In order to use CVS version control with Enterprise Architect, you must install version control software on your local machine. Also, you must create a working directory (using your version control software) before you can configure Enterprise Architect. You can have as many working directories as you like on your local machine.

You must also connect to a repository, which can be either a remote repository 141 or local 145 to your machine. If your repository is local, it must be created with your version control software.

Each working folder you create contains information on connection to a repository. This connection information includes the path to the local or remote repository, the user name and password in order to make a connection.

Note:

To see a video demonstration of setting up a CVS repository for version control, go to http://www.sparxsystems.com.au/resources/demos/settingupCVS/CVS_Final_1.htm.

11.5.3.1 CVS with Remote Repositories

Before you can connect to a remote repository, you must:

- Have version control setup on your local machine
- Have version control setup on a remote server

- Have a working directory on your local machine that points to the repository on the server.

To set up CVS version control with a remote repository, follow the steps below:

1. Ask your system administrator to install CVS and create a remote repository with a module that you can use to control your Enterprise Architect package files. Your administrator must create a username and password for you before you can make a connection.
2. Open a command prompt window and navigate to, or create, a suitable directory to hold your CVS working copy directory; for example:
   C:\> cd myCVSWorkSpace
3. Connect to the remote CVS repository. An example connection command is:
   C:\myCVSWorkSpace> cvs -d:pserver:myUserID@ServerName:/repositoryFolder login

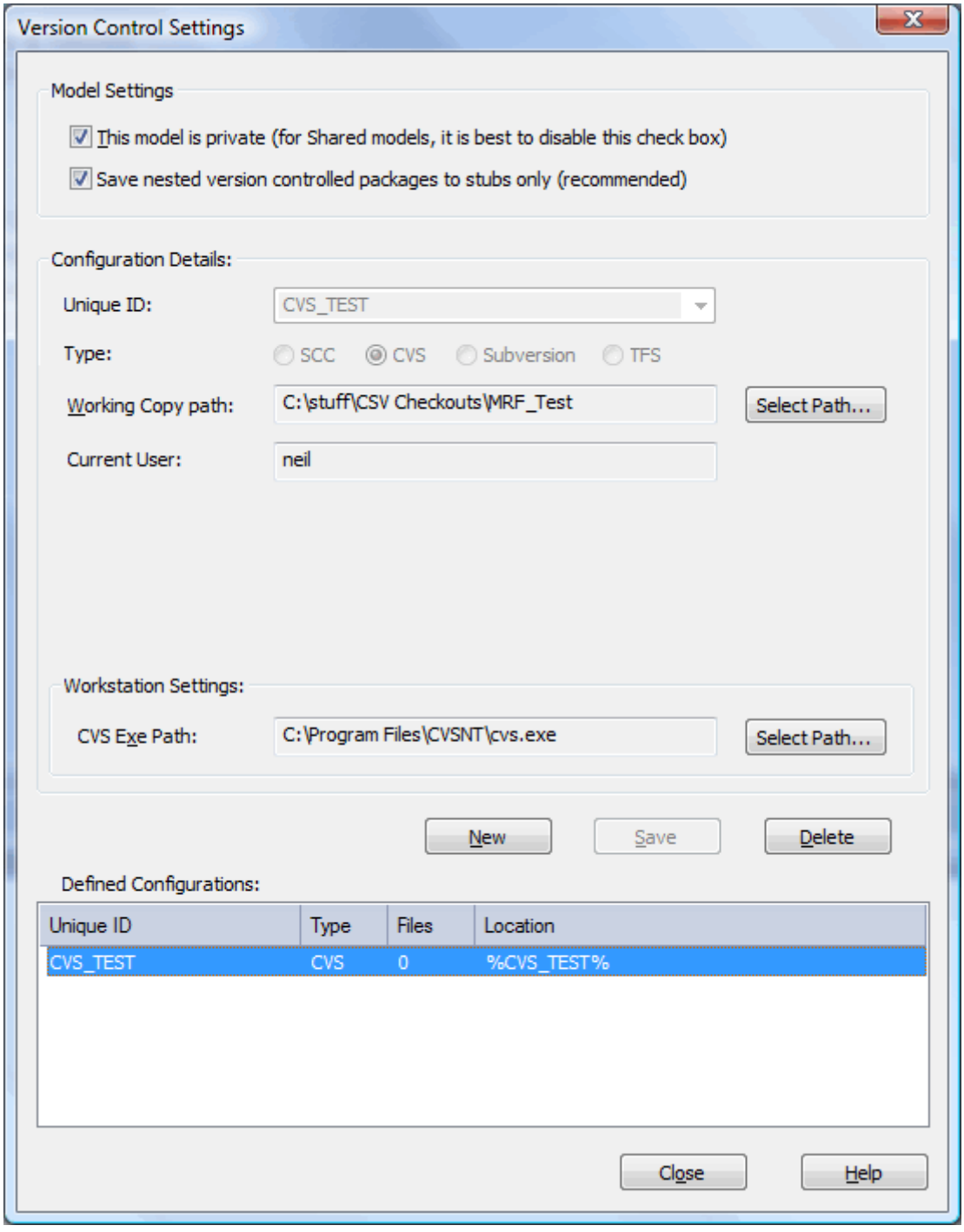
   Note:

   Replace myUserID with your CVS username, replace ServerName with the name of your CVS server and replace repositoryFolder with the path to the repository on the server.
4. Create a local CVS workspace, derived from the remote repository. An example command is:
   C:\myCVSWorkSpace> cvs -d:pserver:myUserID@ServerName:/cvs checkout moduleName

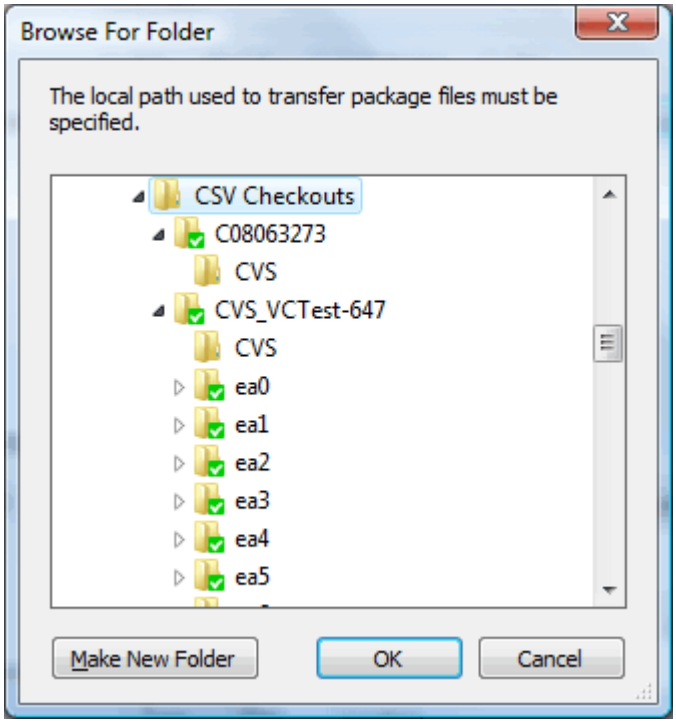
   Note:

   The above command creates a subdirectory in your current working directory, called moduleName. (Replace moduleName with the name of the module created by your system administrator). It creates local copies of all files contained in the CVS module found at ServerName:/cvs.

   It also creates a subdirectory beneath moduleName, called *CVS*. This subdirectory contains a file called Root, that contains your CVS connection information. Enterprise Architect uses this file to obtain your CVS user ID.
5. Verify that your CVS installation is working correctly.
6. Change directory to the one you specified as the working copy, in the cvs checkout command above; that is, C:\myCVSWorkSpace\moduleName
7. Now create a test file, such as **Test.txt**, containing the text *CVS Test*. You can do this with the command:
   echo CVS Test > Test.txt
8. Execute the following CVS commands:
   - cvs add Test.txt
   - cvs commit -m"Commit comment" Test.txt
   - cvs update Test.txt
   - cvs edit Test.txt
   - cvs editors Test.txt
9. The editors command should produce output resembling the following:
   Test1.txt myUserID Tue Aug 9 10:08:43 2009 GMT myComputer C:\myCVSWorkSpace\moduleName
10. Take note of the userID that follows the filename. Enterprise Architect must find and use this user ID when you create your version control configuration. (See the example dialog below.)
11. Launch Enterprise Architect and open or create the model containing the packages to place under version control.
12. Select the **Project | Version Control | Version Control Settings** menu option. The Version Control Settings dialog displays.
13. Click on the **New** button, enter a suitable name in the **Unique ID** 144 field, then click on the **CVS** radio button in the **Type** field.



14. To specify the **Working Copy path** 144 value, click on the **Select Path** button. Select the local folder in which to keep local working copies of the XML files to be stored in the Version Control repository.



15. Click on the **OK** button to return to the Version Control Settings dialog.
16. The **Current User** 145 field should display the user name used to log into the remote CVS repository. If this does not happen, it indicates that Enterprise Architect cannot extract the user name from the file . ..\WorkingCopyPath\CVS\Root and the configuration does not work correctly.
17. If necessary, set the **CVS Exe Path** 145 by clicking on the **Select Path...** button and browsing to the file path for the for file cvs.exe, the CVS executable.
18. Click on the **Save** button to save the configuration you have defined. The new configuration is added to the list of Defined Configurations.

Note:

A new entry is also created in the *Local Paths* list, with the same ID as the new version control configuration. The **Local Path** entry records the Local Project path, for use in subsequent path substitutions.

Options	Use to
This model is Private	Specify whether all users connect to a single shared copy of the model (such as a DBMS) or each user connects to their own private copy of the model. When unselected (for shared models), the option disables the **File History - Retrieve** functionality when the selected package is checked out by another user. This prevents modifications that might have been made by the other user from being discarded through importing a prior revision from version control.
Save nested version controlled packages to stubs only	Set nested version controlled packages to stubs or fully expanded trees. Defaults to selected. For a full explanation of this option, see the *Version Control Nested Packages* 137 topic.
Unique ID	Specify a configuration name that readily distinguishes it from other configurations. The unique ID displays as a selection in a list of Version Control configurations a package can connect to. In addition it is possible to select a previous version control configuration from this drop-down menu providing the configuration is not in use in the current model.
Working Copy path	Specify the folder where the XML files representing the packages are stored.

Options	Use to
	This folder should already exist before it is specified here. Every version control configuration you define in Enterprise Architect, should have its own local working copy folder in which to store working copies of the XMI package files; this should not be a shared network folder. Particularly bear this in mind if you are creating an Enterprise Architect project that is to be shared (e.g. a SQL database).
Current User	Specify the CVS user name associated with all CVS commands that are issued. This name is used by Enterprise Architect, to determine who has a package 'checked-out'.
CVS EXE Path	Specify the full path of the CVS client's executable file.

Note:

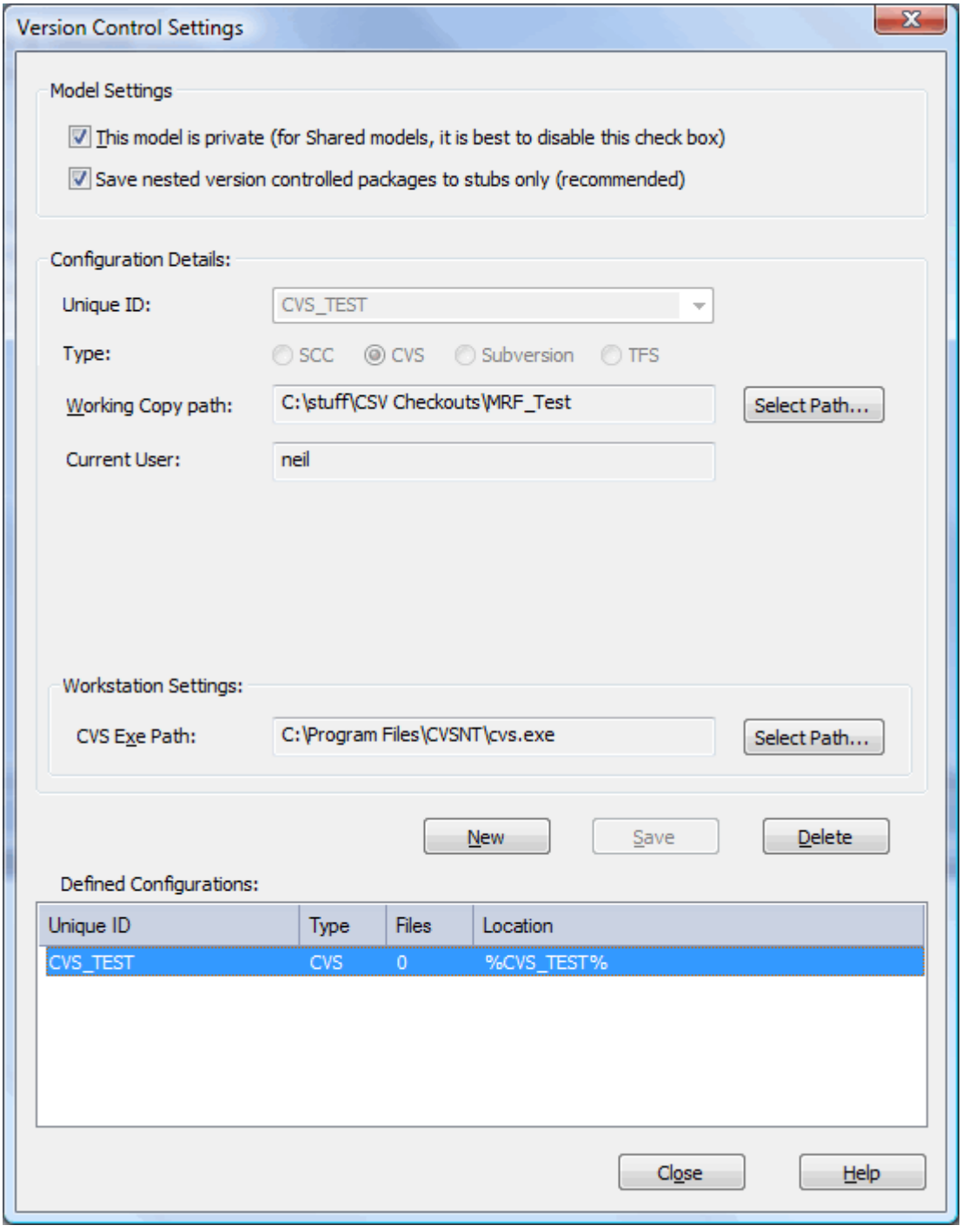
Sparx Systems strongly urge you not to manipulate version controlled package files outside of Enterprise Architect. It is possible to leave the package files in a state that Enterprise Architect cannot recognize.

11.5.3.2 CVS with Local Repositories

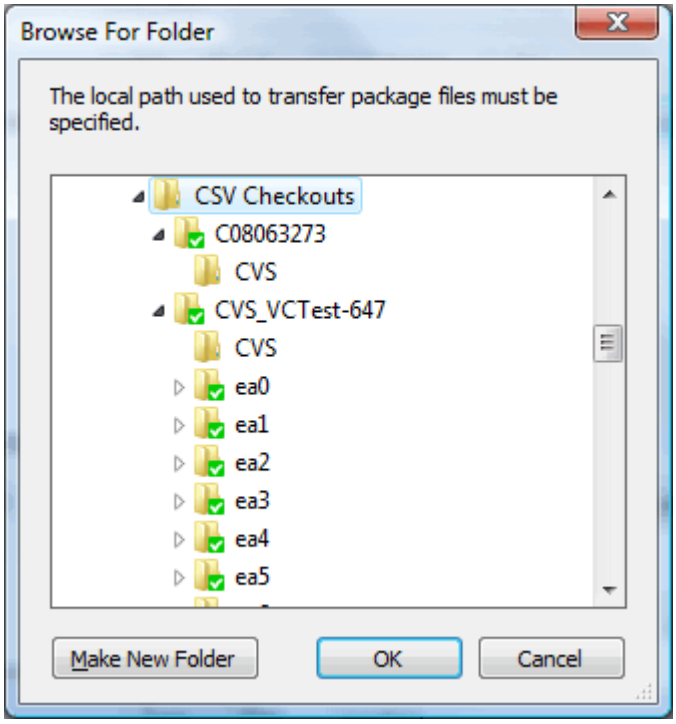
Before you can set up Enterprise Architect, you must have a working directory that points to a local repository; that is, one that is installed on your local machine. See your version control software help files for more information.

To set up CVS version control follow the steps below:

1. Launch Enterprise Architect and open or create the Enterprise Architect model for which packages are to be placed under version control.
2. Select the **Project | Version Control | Version Control Settings** menu option. The Version Control Settings dialog displays.
3. Click on the **New** button.
4. In the **Unique ID** field, type a suitable name for the configuration.
5. Against the **Type** field, click on the **CVS** radio button.



6. Click on the **Select Path...** button to the right of the **Working Copy path** field and browse for and select the local folder in which to keep local working copies of the XML files to be stored in the Version Control repository.
7. If necessary, click on the **Select Path...** button to the right of the **CVS Exe Path** field and browse to the file path for the file *cvs.exe*, the CVS executable.



8. Click on the **Save** button to save the configuration you have defined.
9. The new configuration is added to the list in the Defined Configurations panel.

Note:

A new entry is also created in the Local Paths list, with the same ID as the new version control configuration. The **Local Path** entry records the Local Project path, for use in subsequent path substitutions.

For further information on the fields in the Version Control Settings dialog, see the following table.

Field	Use to
This model is Private	Specify whether all users connect to a single shared copy of the model (e.g. a DBMS) or each user connects to their own private copy of the model. When unselected (for shared models), the option disables the **File History - Retrieve** functionality when the selected package is checked out by another user. This prevents modifications that might have been made by the other user from being discarded through importing a prior revision from version control.
Save nested version controlled packages to stubs only	Set nested version controlled packages to stubs or fully expanded trees. Defaults to selected. For a full explanation of this option, see Version Control Nested Packages 137.
Unique ID	Specify a name that readily distinguishes the configuration from other configurations. The Unique ID is displayed as a selection in the list of Version Control configurations a package can connect to. In addition it is possible to select a previous version control configuration from the drop-down menu providing the configuration is not in use in the current model.
Working Copy path	The folder where the XML files representing the packages are stored. This folder should already exist before it is specified here. Every version control configuration you define in Enterprise Architect, should have its own local Working Copy Folder in which to store working copies of the XMI package files - this should not be a shared network folder. Particularly bear this in mind if you are creating an Enterprise Architect project that is to be shared (for example, a SQL database).

Field	Use to
Current User	The CVS user name associated with all CVS commands that are issued. This name is used by Enterprise Architect, to determine who has a package 'checked-out'.
CVS EXE Path	The full path name of the CVS client's executable file.

Note:

Sparx Systems strongly urge you not to manipulate version controlled package files outside of Enterprise Architect. It is possible to leave the package files in a state that Enterprise Architect cannot recognize.

11.5.4 Version Control with Subversion

Subversion is used to manage files and directories and is an open source version control system. To make use of Subversion control you must have Enterprise Architect version 6.0 or greater.

Tasks in setting up version control with Subversion include:

- Set up Subversion 148
- Create a new Repository Sub-Tree 149
- Create a Local Working Copy 150
- Subversion Under WINE-Crossover 150
- Version Control Configuration 152
- TortoiseSVN 153

Note:

To see a video demonstration of setting up a Subversion repository for version control, go to http://www.sparxsystems.com.au/resources/demos/settingupsubversion/svn_final.htm.

11.5.4.1 Set up Subversion

Obtain and Install Subversion

Note:

Enterprise Architect relies on exclusive file locking when applying version control to its packages. File locking was not introduced into Subversion until version 1.2. Enterprise Architect does not work with Subversion releases earlier than Subversion 1.2.

Before Enterprise Architect can be used with Subversion, the appropriate software must be installed by a Subversion administrator. Ask your system administrator to obtain and install the Subversion server and client applications.

Enterprise Architect must use the Subversion command line client to communicate with the Subversion server; it cannot use other clients such as TortoiseSVN 153.

Important:

Before you attempt to use Subversion through Enterprise Architect, you must first verify that you can use the Subversion command line client to access and operate on files within the working copy folder that Enterprise Architect will use. Your environment must be set up such that you can perform these operations without ever being prompted for user ID or password. For further information, please see the topic *Caching Client Credentials* in the official Subversion documentation.

The official Subversion documentation can be found at: http://www.svnbook.red-bean.com/en/1.4/index.html, while executable files for Subversion can be obtained from: http://www.subversion.tigris.org/project_packages.html#binary-packages.

You require the Windows executables for your client machines running Enterprise Architect in the windows environment. If you plan to run your Subversion server on a non-windows platform, you must download a binary suitable for that platform as well.

Chapter 6 in the Subversion documentation provides guidance on how to configure the server for different methods of access by the client. Secure connection methods are also covered in this chapter.

Your administrator should set up user IDs and passwords for every person who is to access the repository. Your administrator should then provide all users with the path to the repository, and ensure that they can all connect.

Before users can make use of Subversion, they must create local working copies 150 from the repository by checking-out a repository sub-tree 149.

Steps for setting up a repository and creating a local working copy can be found at: http://www.svnbook.red-bean.com/en/1.4/svn.basic.in-action.html#svn.advanced.reposurls.

Note:

Sparx Systems recommend that each new Enterprise Architect model being added to version control with Subversion should have a separate repository sub-tree created for it, and users should create a new local working copy from the sub-tree to be used with that model.

Repository URLs

Subversion repositories can be accessed using many different methods, on local disk or through various network protocols. A repository location, however, is always a URL. The table below describes how different URL schemas map to the available access methods.

Schema	Access Method
file:///	Direct repository access (on local disk).
http://	Access via WebDAV protocol to a Subversion-aware Apache server.
https://	Same as http://, but with SSL encryption.
svn://	Access via custom protocol to an svnserve server.
svn+ssh://	Same as svn://, but through an SSH tunnel.

For more information on how Subversion parses URLs, see http://www.svnbook.red-bean.com/en/1.4/svn.basic.in-action.html#svn.advanced.reposurls.

See Also

- Configure Version Control with Subversion 152

11.5.4.2 Create a new Repository Sub-tree

If a repository sub-tree has already been created for your Enterprise Architect model, skip this topic and see the *Create a Local Working Copy* 150 topic. If your Enterprise Architect model has not previously been added to version control, create a sub-tree for it in your SVN repository by following the steps below:

1. Create a temporary directory structure to import into the SVN repository, which initializes the repository sub-tree for this Enterprise Architect model. The directory structure should look like this:

```
tempDir
|
+--<EA_Model_Name>
   |
   +--trunk
   |
   +--branches
   |
   +--tags
```

2. Open a command prompt, navigate to *tempDir* and issue the command:

```
svn import. <repositoryURL> --message "A Comment of your choice"
```

Note:

After the import is finished, the original tree is not converted into a working copy. To start working, you must still **svn checkout** a fresh working copy of the tree.

3. Delete the directory *tempDir* and all its contents.

For further information see http://www.svnbook.red-bean.com/en/1.4/svn.reposadmin.basics.html.

11.5.4.3 Create a Local Working Copy

Once you have created a sub-tree in the repository for this model, or if one already exists, you are ready to create the local Working Copy for use with this model. Follow the steps below:

1. Choose a suitable directory on your system, in which to create your Subversion Working Copy. The directory that contains your model's .EAP file is probably a good choice.
2. Open a command line window, navigate to the directory to hold your Working Copy directory and check-out the model's sub-tree from the repository, with the following command:

   ```
   svn checkout  <repositoryURL>/<EA_Model_Name>,
   ```

   where <*EA_Model_Name*> is the directory name that you used in setting up the repository sub-tree above.

After you have created your working copy, you should verify everything is working correctly before you attempt to use it from within Enterprise Architect. You must be able to commit files to the repository, without being prompted for ID or passwords.

Enterprise Architect interacts with Subversion using its command line client. Firstly, create a file in your working copy folder then, from a command prompt, add and commit the file to the repository. Use the following commands:

```
svn add <fileName>
svn commit <fileName> -m"A meaningful comment."
```

Now, update the file from the repository, lock the file, edit it and commit once more. Use the following commands:

```
svn update <fileName>
svn lock <fileName>
```

Then edit and save the file using your preferred editor:

```
svn commit <fileName> -m"A meaningful comment."
```

11.5.4.4 Subversion Under WINE-Crossover

When running Enterprise Architect under WINE/CrossOver, you can use either a Windows-based Subversion client or a Linux-based Subversion client.

If you intend to use the HTTP or HTTPS protocols you must use the Unix-like client, as the Windows client cannot access the libraries necessary to create the required network connections. It is also easier to set up your working copy folder using the native Unix-like Subversion client and then continue using that same client from within Enterprise Architect.

However, to make use of the Unix-like client under CrossOver, you must also download and install a bridging utility called *SVN_gate*, which is available from the CodeWeavers' (CrossOver) web site:

http://www.codeweavers.com/support/wiki/EAsvn

When using the Windows Subversion client, you simply install the Win32 Subversion client under CrossOver. Once you have set up your working copy directory, you are ready to use Subversion with Enterprise Architect. However, setting up your working copy is more difficult when using Win32 Subversion under CrossOver. Because you cannot see any output from Subversion commands run under CrossOver, the best way to run the commands is to create a Windows batch file containing the command to run, and to run that batch file as a

Windows command under CrossOver. (See the example batch file 151 below.)

If you are running directly under WINE, launch the Windows batch file from a Unix shell script such as follows;

```
/home/user/cxoffice/bin/wine --bottle "ea" --untrusted
--workdir "/home/user/.cxoffice/ea""/drive_c"
-- "/home/user/.cxoffice/ea/drive_c/batfile.bat"
```

Enterprise Architect uses the Subversion command line client to communicate with your Subversion server. In order for Enterprise Architect to work successfully with Subversion, your Subversion working copy environment must be set up such that you can issue commands to Subversion from the command line, without ever being prompted for user input such as username or password.

By default, whenever the Subversion command-line client successfully responds to a server's authentication challenge, it saves the credentials in the user's private runtime configuration area, which is:

- ~/.subversion/auth/ on Unix-like systems or
- %APPDATA%/Subversion/auth/ on Windows (which translates to .../drive_c/windows/profiles/crossover/Application Data/Subversion/auth/ under CrossOver).

For this reason, Sparx Systems recommend that when you checkout a working copy from the Subversion repository, you specify your username and password on the command line, such that your credentials are cached from the outset. Use a Subversion command such as:

- Using a Unix-like client (run the command from the command line):

```
svn checkout --username "UserName" --password "myPassword" "svn://myServerName:3690/myProject"
"/.../drive_c/workingCopyDirectory"
```

- Using a Win32 client (run the command within a batch file run under CrossOver):

```
"C:\Program Files\Subversion\bin\svn.exe" checkout --username "UserName" --password "myPassword"
"svn://myServerName:3690/myProject" "C:\SVN-test\workcopy" >"C:\SVN-test\stdout.txt" &2>"C:\SVN-test\stderr.txt"
```

It is a good idea to checkout your Subversion working copy into a folder within the WINE bottle where Enterprise Architect is to run. In this way, the pathnames used for your version controlled package files are much shorter.

If you intend to use the Win32 Subversion client with Enterprise Architect, you should create the local working copy that Enterprise Architect is to use, by performing a Subversion checkout command using the Win32 client under WINE/CrossOver. Similarly, if you plan to use the Unix-like Subversion client with Enterprise Architect, you should perform the initial checkout using that client. In this way, your user credentials are cached in the correct location for the client that Enterprise Architect is using.

It is important to verify that your command line client for Subversion is working correctly before attempting to connect from Enterprise Architect. For guidance on verifying your set up, see the *Create a Local Working Copy* 150 topic.

The following is an example of a Windows batch file that can be used under CrossOver to run Subversion commands. Simply uncomment the command to execute. Each command should be a single line - the '\' is intended as a continuation character.

```
rem  "C:\Program Files\Subversion\bin\svn.exe" checkout --username "UserName" --password "myPassword"  \
       "svn://myServerName:3690/myProject" "C:\SVN-test\workcopy"
       >"C:\SVN-test\stdout.txt" &2>"C:\SVN-test\stderr.txt"

rem  "C:\Program Files\Subversion\bin\svn.exe" add "C:\SVN-test\workcopy\myTestFile.xml"   \
       >"C:\SVN-test\stdout.txt" &2>"C:\SVN-test\stderr.txt"

rem  "C:\Program Files\Subversion\bin\svn.exe" commit -m"a message"  "C:\SVN-test\workcopy\myTestFile.xml"   \
       >"C:\SVN-test\stdout.txt" &2>"C:\SVN-test\stderr.txt"

rem  "C:\Program Files\Subversion\bin\svn.exe" lock     "C:\SVN-test\workcopy\myTestFile.xml"   \
       >"C:\SVN-test\stdout.txt"  &2>"C:\SVN-test\stderr.txt"
```

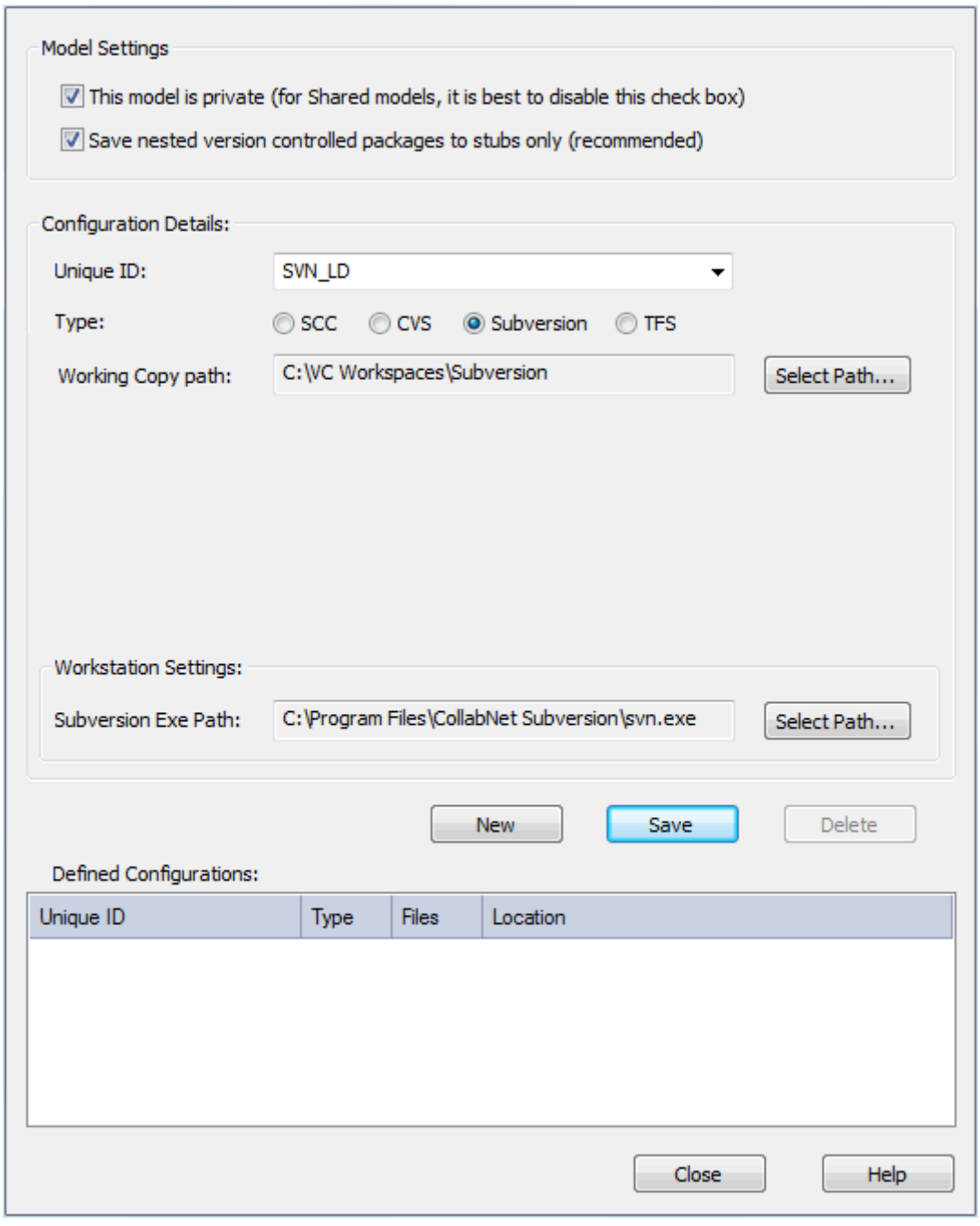
11.5.4.5 Version Control Configuration

This topic assumes that you have already installed Subversion (both the server and the client parts), and that you have a local working copy 150, derived from a repository sub-tree 149, already set up for use with your Enterprise Architect model. If this is not the case, please see the *Set up Subversion* 148 topic.

Once you have set up and tested the Local Working Copy, you are ready to define a Version Control configuration for use with the Enterprise Architect model to place under version control.

To apply version control to your Enterprise Architect model using the Subversion working copy that you have set up, follow the steps below:

1. Launch Enterprise Architect and open the model for which this Working Copy was created.
2. Select the **Project | Version Control | Version Control Settings** menu option.
3. Click on the **New** button, enter a suitable name in the **Unique ID** field, then click on the **Type: Subversion** radio button.



4. To the right of the **Working Copy path** field, click on the **Select Path** button and select the local folder in which to keep local working copies of the XML files to be stored in the Version Control repository.
5. In the Workstation Settings panel, click on the **Select Path** button to specify the path for your Subversion client executable.
6. Click on the **Save** button to save the configuration you have defined; the new configuration is added to the Defined Configurations list.

Note:

A new entry is also created in the Local Paths list, with the same ID as the new version control configuration. The **Local Path** entry records the Local Project path, for use in subsequent path substitutions.

7. When you have finished defining your version control configurations, click on the **Close** button.

Additional Information on the dialog fields:

Option	Use to
This model is private	Specify whether all users connect to a single shared copy of the model (such as a DBMS) or each user connects to their own private copy of the model. When unselected (for shared models), the option disables the **File History - Retrieve** functionality when the selected package is checked out by another user. This prevents modifications that might have been made by the other user from being discarded through importing a prior revision from version control.
Save nested version controlled packages to stubs only	Set nested version controlled packages to stubs or fully expanded trees. Defaults to selected. For a full explanation of this option, see the *Using Nested Version Control Packages* 137 topic.
Unique ID	Specify a configuration name that readily distinguish this configuration from other configurations. The Unique ID displays as a selection in the list of Version Control configurations a package can connect to. In addition you can select a previous version control configuration from this drop-down menu, providing the configuration is not in the current model.
Working Copy path	Specify the folder where the XML files representing the packages are stored. This folder should already exist before it is specified here. Every PC using Subversion version control should have its own Subversion working copy folder in which to store working copies of the XMI package files; this should not be a shared network folder. Particularly bear this in mind if you are creating a .EAP file that is to be shared (for example, a SQL database).
Subversion Exe Path	Specify the full path name of the Subversion client executable file.

Note:

Sparx Systems strongly urge you not to manipulate version controlled package files outside of Enterprise Architect. It is possible to leave the package files in a state that Enterprise Architect cannot recognize.

11.5.4.6 TortoiseSVN

TortoiseSVN is a Windows shell extension for Subversion.

Enterprise Architect cannot use TortoiseSVN to communicate with the Subversion server; it must use the Subversion command line client.

TortoiseSVN provides icon overlays in Windows Explorer that are useful as a tool for observing the status of your Subversion controlled files. It enables you to create your repository sub-trees and check out local working copies from within Windows Explorer using simple menu commands.

Note:

- Sparx Systems recommend that you test your local working copies, by adding and committing a dummy file from the command prompt window.
- Manipulating Enterprise Architect's package files, using tools that are external to Enterprise Architect, could leave those files in a state that Enterprise Architect cannot use.

You can download TortoiseSVN from: http://www.tortoisesvn.tigris.org/.

11.5.5 Version Control with TFS

In order to use Team Foundation Server (TFS) for version control with Enterprise Architect, all users must have either the TFS command line client (*tf.exe*) or Microsoft's Team Foundation Server MSSCCI installed on their local machine. Each intended user must also have an account that provides read/write access to a workspace on the server.

This topic covers configuring version control using the TFS command line client. To configure version control with the TFS MSSCCI client, please follow the instructions in the *Version Control with SCC* 137 topic.

The following preliminary steps should be performed within TFS on each PC and for each user, before making any attempt to define a Version Control Configuration within Enterprise Architect 154 that uses TFS.

Each user must set up a separate workspace for use in version control in Enterprise Architect, containing a single local working folder on their own machine that is mapped to a Source Control folder on the server.

When initializing the connection to TFS, Enterprise Architect issues the command *tf get.* If the *Local Working Copy* path specified in Enterprise Architect's version control configuration is mapped through a TFS workspace that also maps many *other* working folders to their corresponding Source Control folders, TFS can take a long time as it proceeds to update the files in all of those folders. Enterprise Architect might appear to freeze when it initializes the connection to TFS, whilst it waits for TFS to complete the *tf_get* command and hand back program control.

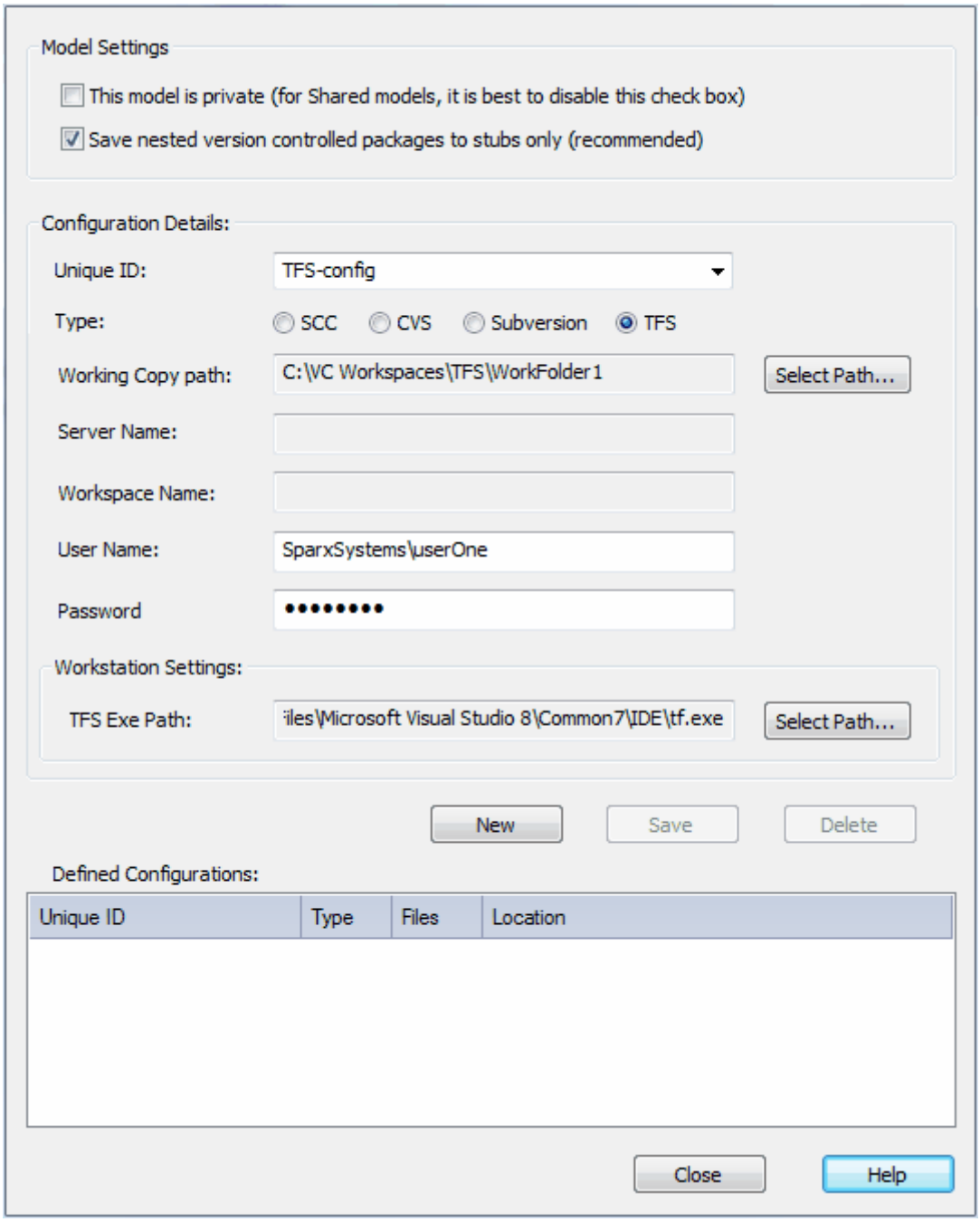
Note:

To see a video demonstration of setting up a TFS project for version control, go to http://www.sparxsystems.com.au/resources/demos/settinguptfs/TFS%20Project-Workspace%20Setup.htm.

11.5.5.1 Connect an Enterprise Architect Model to Version Control using TFS

Having set up TFS and created or otherwise opened your model, you can configure the model for version control under TFS. To do this, follow the steps below:

1. Open or create the Enterprise Architect model to place under version control.
2. Select the **Project | Version Control | Version Control Settings** menu option. The Version Control Settings dialog displays.
3. Click on the **New** button, in the **Unique ID** field enter a suitable name, then select the **TFS** radio button.



4. Click on the **Select Path...** button to the right of the **Working Copy path** field, and select the local folder in which to keep local working copies of the XML files to be stored in the Version Control repository.

Note:

Enterprise Architect queries TFS to retrieve the Server and Workspace names associated with this folder, when attempting to save the configuration data.

5. In the **User Name** and **Password** fields, type values that enable access to the TFS workspace associated with the Working Copy path specified above.

Note:

Users who automatically log in to TFS through means external to Enterprise Architect (for example, through MS Integrated Security) can leave the **User Name** and **Password** fields blank. If the **Password** field is blank, Enterprise Architect retrieves the current user's Windows username and uses that value when determining whether a package is checked out to them or to some other user.

6. The **TFS Exe Path** field displays the default installation path. Click on the **Select Path...** button if it is necessary to modify this field.
7. Click on the **Save** button to save the configuration you have defined.
8. The new configuration is added to the list in the Defined Configurations panel.

Note:

A new entry is also created in the Local Paths list, with the same ID as the new version control configuration. The **Local Path** entry records the Local Project path, for use in subsequent path substitutions.

9. When you have finished defining your version control configurations, click on the **Close** button.

Additional Information on the Dialog Fields

Option	Use to
This model is private	Specify whether all users connect to a single shared copy of the model (such as a DBMS) or each user connects to their own private copy of the model. When unselected (for shared models), the option disables the **File History - Retrieve** functionality when the selected package is checked out by another user. This prevents modifications that might have been made by the other user from being discarded through importing a prior revision from version control.
Save nested version controlled packages to stubs only	Set nested version controlled packages to stubs or fully expanded trees. Defaults to selected. For a full explanation of this option, see Use Nested Version Control Packages 137.
Unique ID	Specify a configuration name that readily distinguishes it from other configurations. The Unique ID is added to the list of Version Control configurations a package can connect to. In addition it is possible to select a previous version control configuration from this drop-down menu providing the configuration is not in the current model.
Working Copy Path	Specify the folder where the XML files representing the packages are stored. This folder should already exist before it is specified. Every PC using TFS version control should have its own TFS Local Folder in which to store working copies of the XMI package files - this should not be a shared network folder. Particularly bear this in mind if you are creating a .EAP file which is to be shared (for example, a SQL database).
Server Name	Specify the name of the Team Foundation Server to connect to.
Workspace Name	Specify the name of a pre-defined TFS workspace that you are using.
User Name	Specify the user name that you use to connect to the Team Foundation Server. The user name that you specify should give you read/write permissions in the specified workspace.
Password	Specify the password associated with the user name you specify. Enterprise Architect stores this password, in encrypted form, as part of the version control configuration data.

Option	Use to
TFS Exe Path	Browse to and select the full path name of the TFS command line client's executable file.

Notes:

- Sparx Systems strongly urge you not to manipulate version controlled package files outside of Enterprise Architect. It is possible to leave the package files in a state that Enterprise Architect cannot recognize.
- Visual Studio Integration (MDG Integration for Visual Studio 2005 or 2008) enhances TFS support by providing access to, for example, work items and bugs within both Enterprise Architect and the MDG Integration product.

11.6 Use Version Control

The following topics describe the most common activities using the version control features of Enterprise Architect, accessed through the Package Version Control Menu 157:

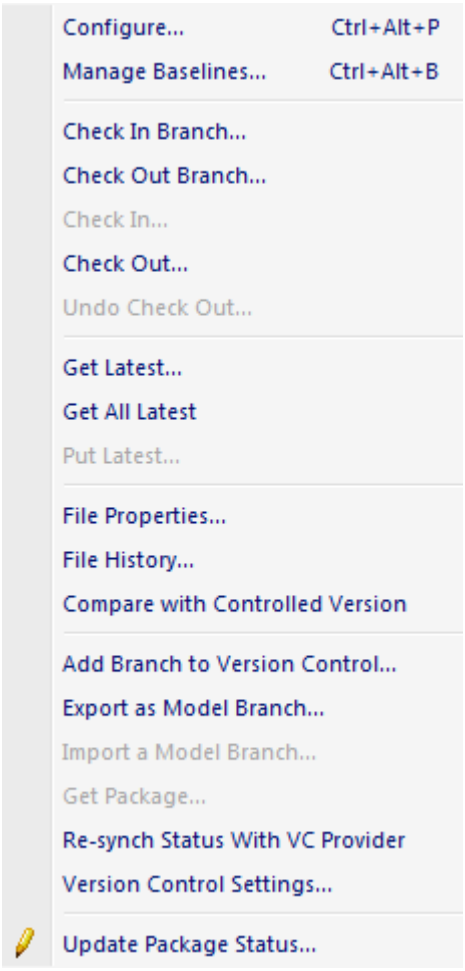
- Configure Controlled Package 160
- Use Existing Configuration 161
- Validate Package Configurations 161 (**Project | Version Control** menu)
- Check In and Check out Packages 162
- Include Other Users Packages 164
- Apply Version Control to Branches 165
- Export Controlled Model Branch 166
- Import Controlled Model Branch 166
- Review Package History 168
- Refresh View of Shared Project 168

General Notes

- The export/import facility is not fast and submitting packages containing many sub-nodes to version control should be avoided. It is recommended version control is applied to individual packages. See the *Version Control Nested Packages* 137 topic for more information.
- Replication 100 should not be combined with version controlled packages.
- Sparx Systems strongly urge you not to manipulate version controlled package files outside of Enterprise Architect. It is possible to leave the package files in a state that Enterprise Architect cannot recognize.

11.6.1 Package Version Control Menu

To display the **Version Control** menu, right-click on a version controlled package in the Project Browser and select the **Package Control** context menu option. (This menu displays a number of different options if the selected package is not under version control - see the *Controlled Package Menu* 110 topic.



Menu Option & Function Keys	Use to
Configure [Ctrl]+[Alt]+[P]	Display the Package Control Options 160 dialog which enables you to specify whether this package (and its children) is controlled, which file it is controlled through, and which version control configuration to use.
Manage Baselines [Ctrl]+[Alt]+[B]	Create a Baseline 208 of the current package, or compare the current package with a previous Baseline.
Check In Branch	For the selected branch of the model, (that is, the selected package and all of its child packages) display the Select Packages to Check In 163 dialog, listing all version controlled packages within that branch that are checked out to you. You can then select packages in the displayed list, to be submitted for check-in.
Check Out Branch	For the selected package, check out the package and recursively check out all of its contained sub-packages 164. Retrieves the latest version of the packages from the central repository, overwriting the current packages. After check out, the packages are available for editing.
Check In	Submit the currently selected package to the central repository. Enterprise Architect prompts you to enter optional comments describing changes to the packages.
Check Out	Retrieve the latest version of the currently selected package from the central repository, overwriting the current packages. After check out the

Menu Option & Function Keys	Use to
	packages are available for editing.
Undo Check Out	Cancel all changes you have made to the currently selected package. This restores the model to the state it was in before the package was checked out, leaving the select package and sub-packages locked.
Get Latest	Retrieve the latest revision of the package from the repository. Available only for packages that are checked in.
Get All Latest	Retrieve the latest revision of all version controlled packages in the project. Only retrieves packages that are checked in.
Put Latest	Update the central repository with the currently selected package (which you have checked out), while retaining checkout status on the package. This is equivalent to checking a package in and immediately checking it back out again.
File Properties	Ask the version control provider to show the version control properties associated with the XML export file pertaining to the currently selected package. This also identifies who has checked out the package.
File History	Where the controlling package has been configured by an SCC provider, this provider shows a change history for the package. See your provider's documentation for details on how to use the control. Otherwise, if the version control is CVS, the history is shown via Enterprise Architect's internal CVS history menu.
Compare with version on disk	Compare the current package with the XMI version on disk.
Add Branch to Version Control	Apply version control to all packages within a selected *model branch*, in a single operation. In this context, a model branch is a package that is currently selected in the Project Browser, and all of the packages contained within it.
Export as Model Branch	Export 166 a newly created model branch from your own private copy of a model.
Import a Model Branch	Retrieve 166 a model branch and import it into either the source model or another model.
Get Package	Access packages in the version control repository that are not currently available in your model.
Re-synch Status With VC Provider	Update the version control status value recorded for the selected package in the Enterprise Architect project to match the value reported by the version control provider 169, without performing an XMI import or export. Use this function when the package's version control status recorded in your Enterprise Architect project is out of synchrony with the version control status reported by your version control provider.
Version Control Settings	Display the Version Control Settings dialog 135.
Update Package Status	Provide a bulk update on the status of a package, including status options such as **Proposed**, **Validate** and **Mandatory**. **Note:** This option is a generic package option not specific to version control.

11.6.2 Configure Controlled Package

Before working on a package under version control, you must define it as a controlled package and specify the version control configuration to use.

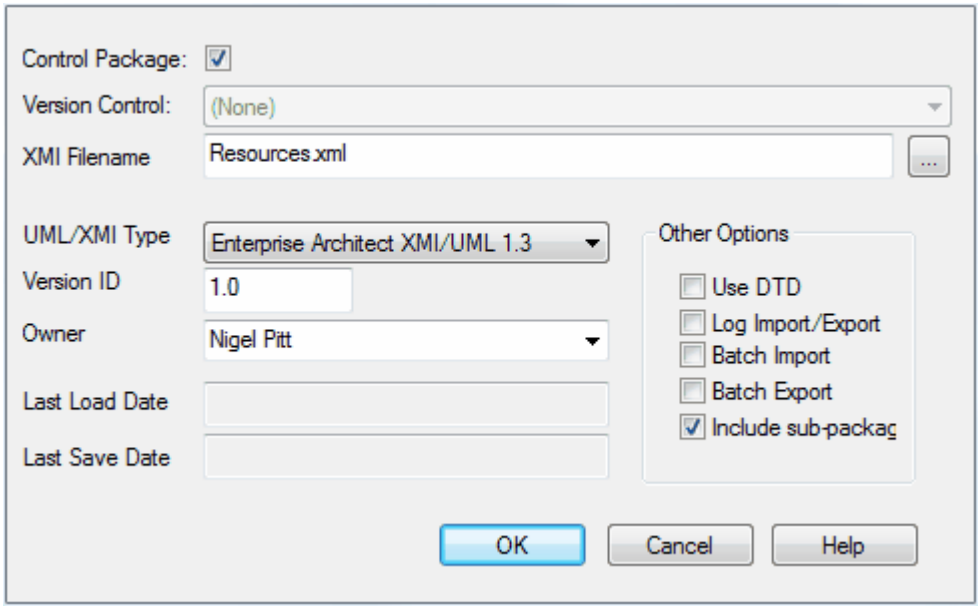
Note:

In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Configure Packages** 183 permission to configure packages for version control.

Configure a Version Controlled Package

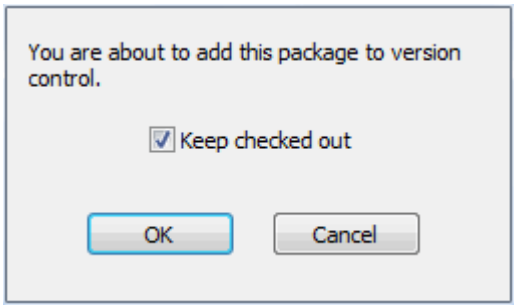
To configure a version controlled package, follow the steps below:

1. In the Project Browser, right-click on the package to place under version control. The context menu displays.
2. Click on the **Package Control | Configure** menu option. The Package Control Options dialog displays.

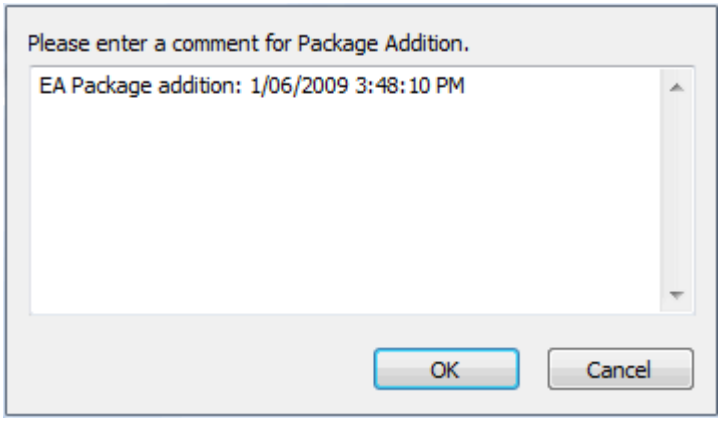


3. Select the **Control Package** checkbox to indicate that this is a controlled package.
4. Click on the **Version Control** drop-down arrow and select the version control repository; this connects the package to a specific version control configuration.

   The **XMI Filename** field then displays the version control configuration default path.
5. The **Version ID** field defaults to **1.0**; if necessary, change this to the appropriate reference.
6. The **Owner** field defaults to your user name; if necessary, type or select the name of the user who owns the package.
7. Click on the **OK** button to set the version control options. The Add Package to Version Control dialog displays.



8. If you do not want to check-out the package immediately, clear the **Keep checked out** checkbox.
9. Click on the **OK** button. The Add Comment window displays.



This window displays the date and time at which the package was put under version control.

10. If required, type any further comments in the window. Click on the **OK** button.

Enterprise Architect places the package under the version control configuration you selected, and marks the package in the Project Browser with the version controlled checked out or not checked out 132 icons, as appropriate.

11.6.3 Use Existing Configuration

Once a version control configuration has been defined in one model it is possible to add the configuration to other models. To use this feature follow the steps below:

1. Open the model that is to have the predefined version control configuration added to it.
2. Right-click on any package in the Project Browser and select the **Package Control | Version Control Settings** context menu option. The Version Control Settings 135 dialog displays.
3. Click on the **New** button.
4. In the **Unique ID** field, click on the drop-down arrow and select one of the previously-defined version control configurations.
5. Click on the **Save** button to confirm the version control configuration.

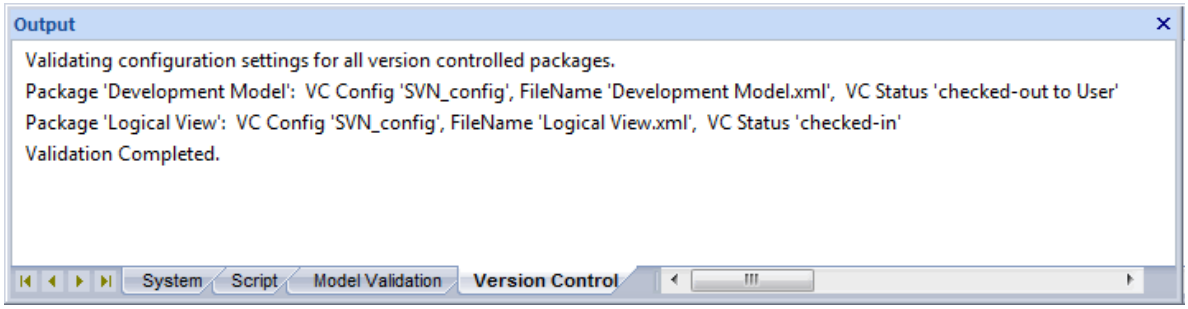
11.6.4 Validate Package Configurations

You can test the validity of the version control settings associated with each version controlled package within your current model. To do this, select:

Project | Version Control | Validate Package Configurations

The validation process scans the model database and verifies that the version control configuration associated with each version controlled package is fully specified in the current model. It also queries the corresponding version control provider to find the status of the package file associated with each version controlled package.

The results of the validation process are sent to the Enterprise Architect Output window, as shown below:



Depending on the results, you can then complete the definition of any invalid or missing version control configurations, or correct problems with individual packages or their associated package files.

Click on an error message to highlight, in the Project Browser, the package that is in error.

11.6.5 Check In and Check Out Packages

To work on a version controlled package you must have the package checked out. When a package is checked out to a specific user, a write lock is set on the package and other users cannot make changes to it until it has been checked in again.

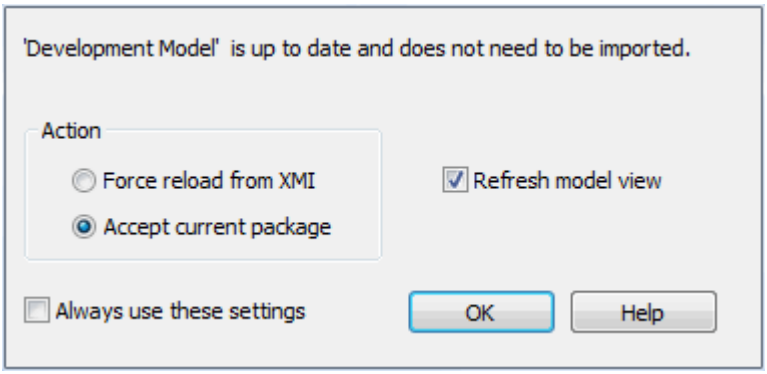
Note:

In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Use Version Control** 183 permission to check files in and out using version control.

Check In/Check Out

1. In the Project Browser, right-click on the package icon.
2. Select the **Package Control | Check In**, **Check Out** or **Undo Checkout** context menu options, as appropriate.
3. If required, enter a comment when prompted to do so.

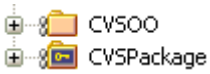
If you are working in a private 131 model and you select the **Check Out** menu option, the Import Package dialog displays (in shared models, this dialog only has default values and therefore does not display).



Option	Use to
Force reload from XMI	Reload the package content from the XMI file in the central repository, even though the package and XMI file are synchronized. This ensures that links and dependencies that might not have been refreshed are updated as well.
Accept current package	(The default.) Leave the package content in its current state.

Option	Use to
Refresh model view	Refresh the model view to show any changes from other checked out packages.
Always apply above settings	Apply the settings in the above three fields every time you check out a package that is found to be up to date, and therefore do not display this dialog again. **Note:** To display the dialog if, for example, you want to change the settings, press **[Ctrl]** while you select the **Package Control | Check Out** menu option.

The package icon in the Project Browser should change. When you check out a package this is represented by a figure 8 to the left of the package icon. When you check in a package the package icon is overlaid with a colored rectangle and key. In the example below, the upper package is checked out whilst the lower package is checked in.

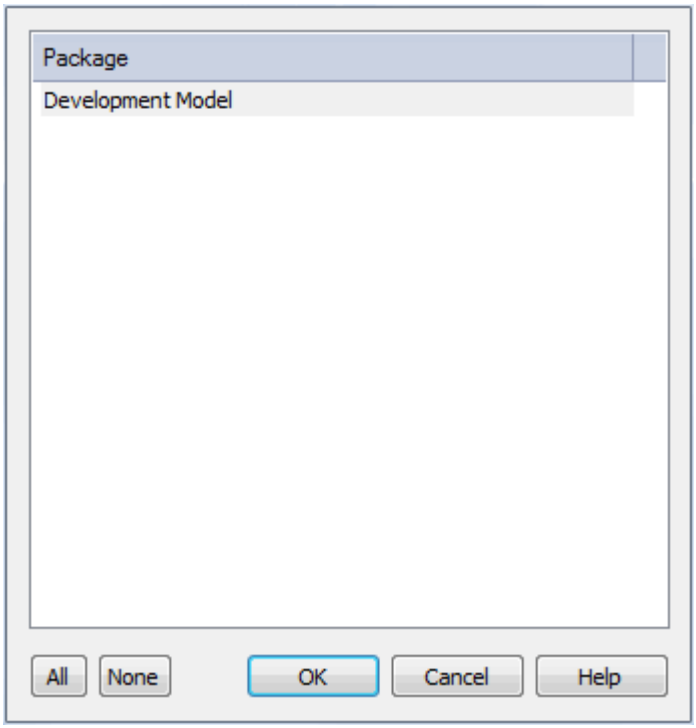


Notes:

- If you check out a version controlled package whilst offline, the package icon has a red figure 8 in front of it. See Offline Version Control 169.
- If the packages under version control contain any alternative images **and** those images are subject to **frequent change**, you can set the **Export alternate images** option on the Options dialogto export the images to the version control repository when you check in the packages. If the images are not subject to frequent change, do not select this option and instead use Export/Import Reference Data 250 to manage alternative images.

Check In Branch

1. In the Project Browser, right-click on the package icon at the root of the model branch that is to be checked in and select the **Package Control | Check In Branch** context menu option. The Select Packages to Check-in dialog displays, listing all version controlled packages within the branch that you have checked out.



2. Click on the package to check in, or use:
   - **[Ctrl]+click** to add or remove several individual packages
   - **[Shift]+click** to select a range of packages
   - **All** to select all packages listed
   - **None** to clear all selected packages.
3. Click on the **OK** button to check-in the selected packages.
4. If required, enter a comment when prompted to do so. (This comment applies to all packages that you have checked in.)
5. Each package icon changes to indicate that the packages have been checked-in.

Check Out Branch

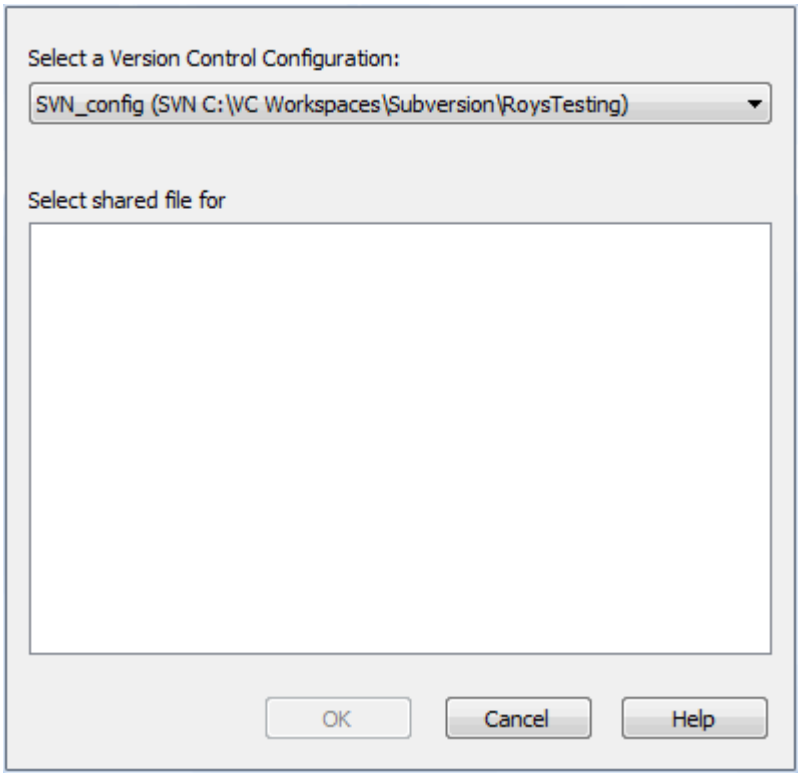
1. In the Project Browser, right-click on the name of the root package of the model branch to be checked out and select the **Package Control | Check Out Branch** context menu option. The selected package and all of its contained sub-packages are recursively checked out.
2. Any packages that cannot be checked-out are listed in a message box, with a brief description of the problem. For example: *The package is already checked out by user 'Fred'*.
3. When Project Security is enabled in *Lock to Edit* mode, Enterprise Architect prompts you to apply a User Lock throughout the selected model branch before proceeding.

11.6.6 Include Other Users' Packages

You can retrieve packages that have been created by other users, or by you in another model, from version control and import them into your current model.

Other users might be creating packages to use in your model. If you are not sharing a SQL database or .EAP file, those packages do not automatically become part of your model. If the packages have been placed into version control, you can retrieve them and import them into your model as children of an existing package, using the **Get Package** command.

1. You must have access to the package files through the version control system and you must define a Version Control Configuration through which to access those files. The version control configuration must use the same unique ID that was originally used to add the package to version control.
2. In the Project Browser, right-click on the package to use as the parent of the incoming package.
3. Select the **Package Control | Get Package** context menu option. The Get Shared File dialog displays.



4. In the **Select a Version Control Configuration** field, click on the drop-down arrow and select the version control configuration associated with the package to retrieve. The file list is populated with the names of files available through that configuration, for retrieval and import into your model.
5. Select the package file to import into your model and click on the **OK** button.

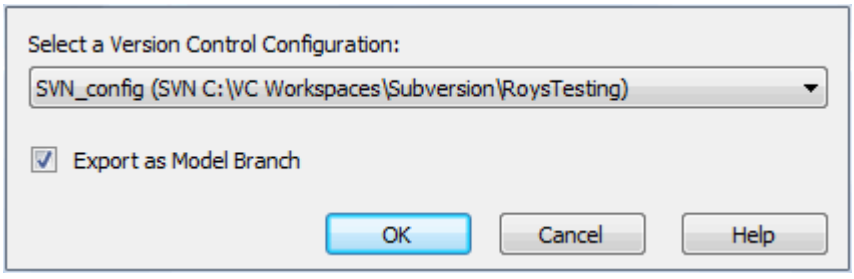
11.6.7 Apply Version Control To Branches

Enterprise Architect enables you to apply version control to all packages within a selected *model branch*, in a single operation. In this context, a model branch is a package that is currently selected in the Project Browser, and all of the packages contained within it.

The Version Control Configuration 135 to be used in this operation must be defined within the model before selecting this command.

To apply version control to a model branch, follow the steps below:

1. Right-click on the required package and select the **Add Branch to Version Control** 157 context menu option. The Apply VC to Branch dialog displays.



2. In the **Select a Version Control Configuration** field, click on the drop-down arrow and select the Version Control Configuration to use.
3. If required, select the **Export as Model Branch** checkbox to export the selected package (and sub-packages) as a Model Branch 166.

4. Click on the **OK** button. Enterprise Architect creates a number of sub-folders within the version control working copy folder, before exporting all of the packages within the selected model branch. Enterprise Architect generates package filenames using the package GUIDs, before adding the resulting files to version control.

If you have selected the **Export as Model Branch** checkbox, once the version control operation is complete Enterprise Architect also creates a Model Branch file (.EAB file). You can subsequently import the version-controlled Model Branch 166.

11.6.8 Export Controlled Model Branch

You might want to export a newly created model branch from your own private copy of a model so that, for example:

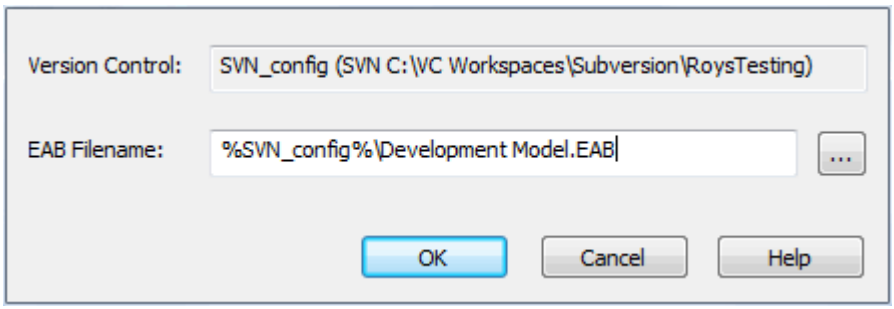
- Another user can import that branch into their own private copy of the same model, or
- It can be imported for inclusion as a common branch in a number of different models.

Applying version control to an Enterprise Architect model can result in many XMI files placed under version control. It could then be hard to locate and import the file corresponding to the root of a particular model branch. Enterprise Architect's Model Branch Files (.EAB files) overcome this problem by simplifying the retrieval of model hierarchies for use in other models.

The facility is only enabled for packages that are already under version control. The exported Model Branch File is also placed under version control, using the same version control configuration that is controlling the selected package.

To export a model branch, follow the steps below:

1. Right-click on the version controlled package to export as a model branch.
2. Select the **Package Control | Export as Model Branch** context menu option. The Export as Model Branch dialog displays.



3. In the **EAB Filename** field, type a name for your Model Branch File. Alternatively, click on **[...]** and browse for the file location.

Note that the package name is supplied as a default.

You can specify any file name, including sub-folder names, as long as the file is contained in or below the working folder of your version control configuration.

11.6.9 Import Controlled Model Branch

It might be necessary to either:

- Retrieve a model branch created by another user in a private copy of a model, to import it into your own private copy of the same model or
- Retrieve a model branch that is common in many models, for inclusion in a new model.

Applying version control to an Enterprise Architect model can result in many XMI files placed under version control. It could then be hard to locate and import the file corresponding to the root of a particular model branch. Enterprise Architect's Model Branch files overcome this problem by simplifying the retrieval of model hierarchies for use in other models.

The **Import a Model Branch** context menu option uses Enterprise Architect's Model Branch Files, of which there are few, to retrieve information about the root package file and import the model branch. The Model

Branch File records information such as the name and type of the version control configuration for the selected package, and the relative filename of the version controlled XMI file associated with the package.

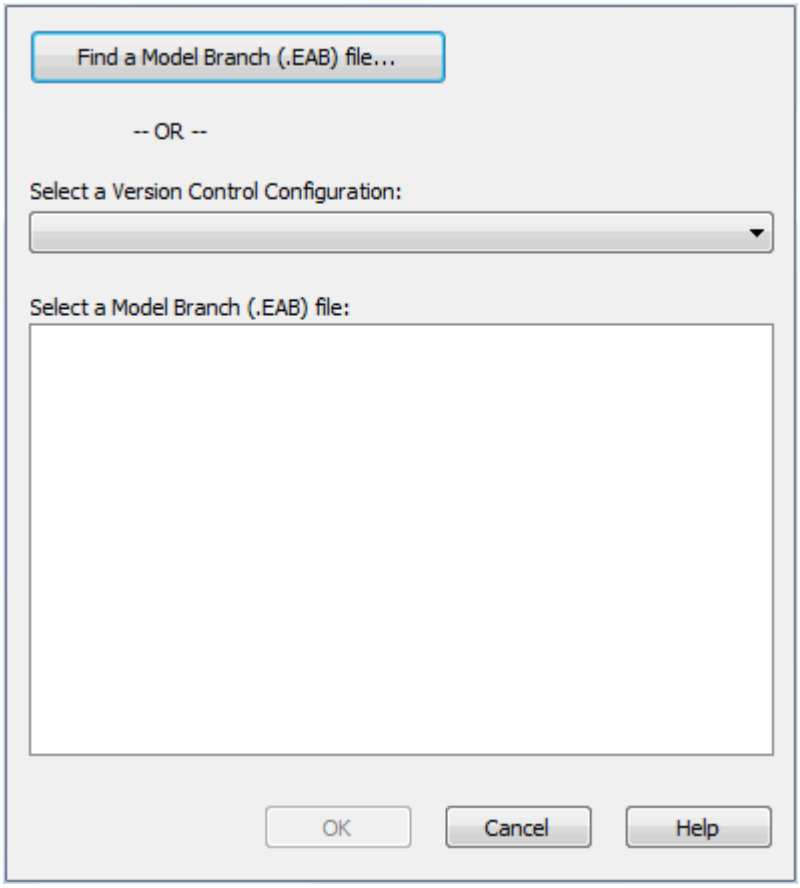
Before attempting to import a model branch, you must have access to the version controlled XMI files that represent the model branch to be imported. That is, there must be a working copy folder, accessible from the machine on which Enterprise Architect is running, that is associated with the Version Control repository containing those XMI files.

It is not necessary to have the relevant Version Control Configuration set up within Enterprise Architect before issuing this command - Enterprise Architect prompts you to complete specification of the configuration if necessary.

The **Import a Model Branch** context menu option is only enabled for packages that you (the current user) are able to edit, as the imported model branch is inserted into the model under the selected package.

To import a model branch, follow the steps below:

1. Right-click on the package into which the model branch is to be imported.
2. Select the **Package Control | Import a Model Branch** context menu option. The Import VC Model Branch dialog displays.



3. Either:
   - Click on the **Find a Model Branch (.EAB) file** button and browse for the Model Branch File. If the version control configuration used by the file has not been fully set up, Enterprise Architect prompts you to complete and save the configuration. The model branch import then proceeds. OR
   - If the version control configuration used by the file has been fully set up in the current model, click on the drop-down arrow in the **Select a Version Control Configuration** field and select the configuration, then select the Model Branch File from the **Select a Model Branch (.EAB)** file list. Click on the **OK** button to import the model branch.

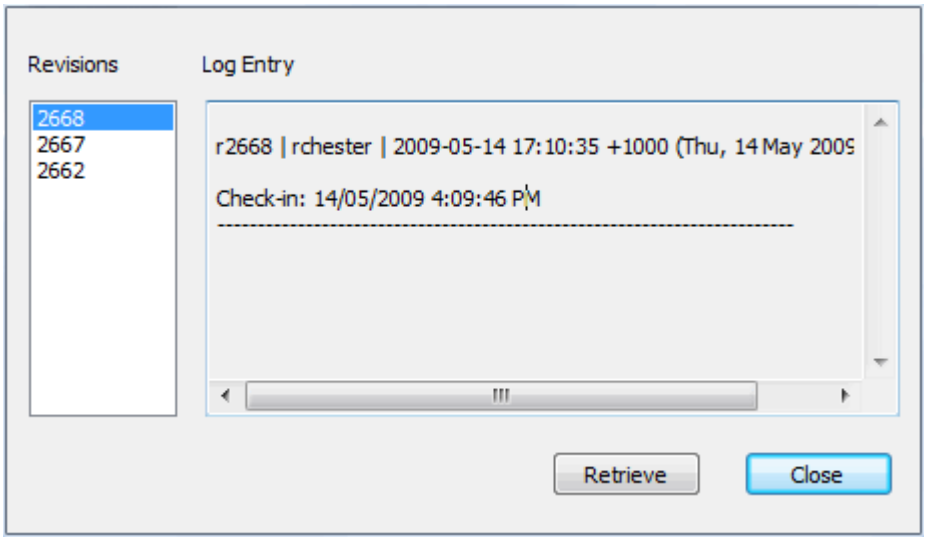
Enterprise Architect imports the root package specified in the Model Branch File and recursively imports and populates all the sub-packages contained in the root package.

11.6.10 Review Package History

Reviewing package history enables you to view the history of checked in package revisions. It also enables you to import selected prior revisions of the package into your model. The package revision is retrieved in read-only form, enabling you to view the package contents but not make any changes to the package.

To review package history follow the steps below:

1. In the Project Browser, right-click on the package configured for version control. The context menu displays.
2. Select the **Package Control | File History** menu option.
   - If the package has been configured through SCC, you access the history through the mechanism offered by the third party SCC provider.
   - If the package has been configured for CVS, Subversion or TFS, Enterprise Architect's File Version History dialog displays and the following steps apply.



3. In the **Revisions** field, click on a revision number to view the log entries for that revision.
4. To view the package history select a revision and then click on the **Retrieve** button. A warning dialog displays, indicating that the package is being opened in read-only mode.
5. Click on the **Yes** button to continue or the **No** button to cancel the action. The package is retrieved in read only mode, enabling you to view the history of the package at the specified version.
6. To go back to the latest version, in the Project Browser right-click on the package and select either the **Package Control | Check Out..** context menu option or the **Package Control | Get Latest** context menu option.

11.6.11 Refresh View of Shared Project

When a user of a shared model checks out a package and makes changes, other users can see those changes by refreshing their view of the package or the changed diagram within the package.

You can refresh your view of the Project Browser in the following ways:

- Right-click on the package name in the Project Browser and select the **Contents | Reload Current Package** context menu option
- Select the **File | Reload Current Project** menu option (or select the **Reload Project** icon in the Project toolbar, or press **[Ctrl]+[Shift]+[F11]**)
- Close the project and reopen it.

You can refresh the current diagram in the following ways:

- Select the **Window | Reload Current View** menu option
- Right-click on the opened diagram tab in the diagram view, and select the **Reload <diagram name>** context menu option.

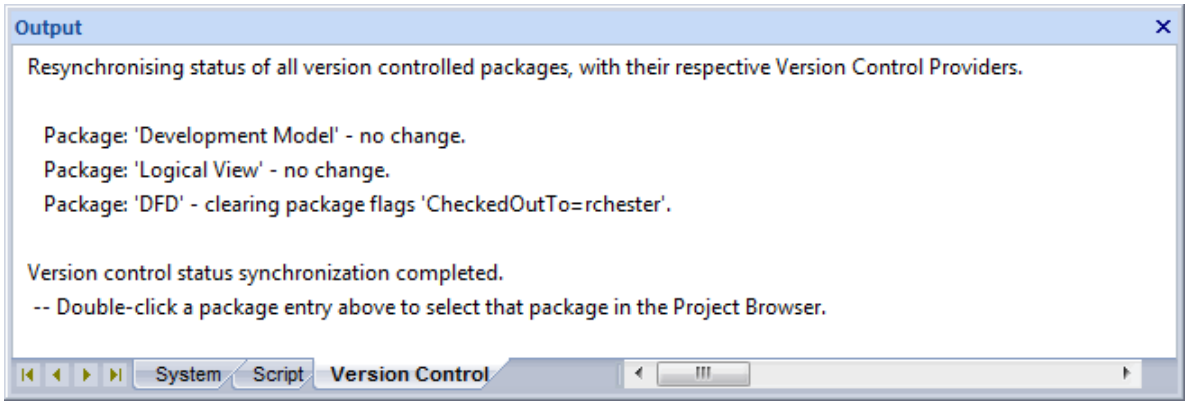
11.6.12 Resynchronize the Status of Version Controlled Packages

Enterprise Architect enables you to re-synchronize the version control status of either a single version controlled package or all version controlled packages within your current model with the status reported by your version control provider. This can be useful when you give a copy of a .EAP file configured for version control to a new team member.

- For a single package, right-click on the package name in the Project Browser and select the **Package Control | Re-synch Status With VC Provider** menu option.
- For all version controlled packages within the project, click on the **Project | Version Control | Re-synch Statuses of All Packages** menu option.

For a given package, the re-synchronization process queries the corresponding version control provider to find the status of the package file associated with the version-controlled package. If necessary, the process then updates the package flags within the model database, to synchronize the package status recorded in the model with the value reported by the version control provider.

The results of the re-synchronization process are sent to the Enterprise Architect Output window, as shown below:



Double-click on any result message to select, in the Project Browser, the corresponding package.

Note:

This process does not cause any package data to be:

- exported from your model to the associated package file, or
- imported from a package file into your model's package data.

If a package has been checked-out and modified with Enterprise Architect, but your version control provider reports the package file as checked-in, running this process marks the package within Enterprise Architect as being checked-in, without exporting and committing the pending changes. Subsequently checking-out the package imports the latest revision of the package file from version control, effectively discarding the uncommitted modifications from the model.

Similarly, if a package file is checked-out in the version control system, but not in the Enterprise Architect model, running this process marks the package within the model as checked-out, but it does not import the associated package file from the version control system. Consequently, it is possible to check-in a package from Enterprise Architect that is potentially out of date, compared to the latest revision of the package file within the version control system.

11.7 Offline Version Control

When loading a model that uses version control, Enterprise Architect normally initializes a connection to the version control system for each Version Control Configuration defined in the model. If Enterprise Architect is unable to connect a Version Control Configuration for any reason, it displays warning messages to notify you and provides 'offline' version control functionality for all packages associated with the failed connection.

You can prevent Enterprise Architect from attempting to make any version control connections by selecting the **Project | Version Control | Work Offline** menu option before loading a model. This is useful if you know

that Enterprise Architect cannot connect to your version control system. For example, if you are working on a laptop computer that is disconnected from your network and you have an Enterprise Architect model that uses a large number of Version Control Configurations, choosing to work offline before you load the model enables you to avoid all the error messages that Enterprise Architect would normally display as each version control connection attempt fails.

You can switch between working offline and working online at any time, either before or after a model is loaded. Toggle the **Project | Version Control | Work Offline** menu option. Enterprise Architect disconnects or reconnects version control (depending on connection availability) according to your selection.

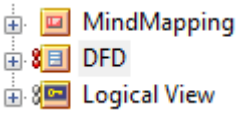
Use Version Control Whilst Disconnected From Your Version Control Server

Enterprise Architect 'remembers' the status of a model's version controlled packages. Packages that were checked out 162 to you prior to disconnecting from the server are still shown as checked out to you, even though you are no longer connected to the server. You can still edit these packages as you normally would.

Packages that were not checked out to you prior to disconnecting from the server are shown as version controlled and locked. You cannot edit these packages until you check them out.

Offline Check Out

In releases of Enterprise Architect from release 6.0 onwards, you can 'check-out' and edit a version controlled package even when your machine is disconnected from the version control server. In the example below, the colored 'figure 8' icon 132 for *DFD* indicates that you have checked it out whilst offline (the gray 'figure 8' icon shown against *Logical View* indicates that you have checked out a version-controlled package online).



Important:

You should be aware that the version control system - and therefore other users - have no way of knowing that you have 'checked-out' a package whilst offline. It is not possible to merge changes to an XMI file that result from two users editing the same package at the same time. If an offline checkout leads to two people editing the same package at the same time, when the changes are brought back online the first-saved set of changes is lost.

Check in a Package That Was Checked Out Offline

Once you reconnect your machine to the version control server, if the package you checked out offline is not currently checked out by another user, you can check in that package. However, before Enterprise Architect checks in such a package, it compares the local working copy of the package file with the latest revision in the repository. (These package files remain unchanged in your work area until Enterprise Architect exports the package again before checking in.) If the repository version remains unchanged from when you last updated your local copy, Enterprise Architect exports and checks in your package without further prompting.

On the other hand, if the repository now contains a file that has changed since you last updated your local copy, checking in your package overwrites whatever those changes might be. Enterprise Architect displays a message warning you of the pending data loss and giving you the opportunity to abort the check in. At this point, you must decide whether to discard your own changes, using the **Undo Check Out** command, or continue with your check in and overwrite the changes that have been committed to the repository since you last updated your local copy from the repository.

You can use the **File Properties** command to determine who checked in the last changes to this package. This might help you to discover what changes have been uploaded and decide whose changes take precedence.

Update Before You Disconnect

Whenever you are connected to the version control server, you are always working with the latest version of a package. This is because you cannot modify a package until you check it out from version control, and checking it out loads the latest revision from the repository into your model.

These rules do not apply when you are disconnected from the version control server. You are working on

whatever versions you have on your machine, dating back to the last time you updated your local copy of each version controlled package. So, if you are planning to work on a model whilst disconnected from version control, it is a very good idea to make sure that you have the latest versions of all packages before you disconnect. The Get All Latest 157 option makes this a simple task.

12 User Security

What is User Security in Enterprise Architect?

User security in Enterprise Architect can be used to limit the access to update functions within the model. Elements can be locked per user or per group. Where user security is enabled a password is required to log in to the model. Security in Enterprise Architect is not designed to prevent unauthorized access; rather it is intended as a means of improving collaborative design and development by preventing concurrent editing and limiting the possibility of inadvertent model changes by users not designated as model authors.

With workflow administration permissions 183, you can also develop workflow scripts 187 (using the Scripter window). Workflow scripts validate and control 188 user input.

User Security Basics

User security is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect. It offers two policies: the standard security mode and the rigorous security mode.

- In the standard security mode all elements are unlocked and, as necessary, a user can set a user or group lock on any element or set of elements in order to make changes and protect those changes.
- Under the rigorous security mode an Enterprise Architect model is read-only and nothing in the model can be edited until explicitly checked out with a user lock.

For more detailed information on the security policies see the *Security Policy* 174 topic.

User Security Tasks

A number of security tasks can only be performed by users with Administrative rights to the model. These tasks include:

- Security Policy 174
- Enable Security 173
- Maintain Users 175
- Import User IDs From Active Directory 177
- Change User Passwords 190
- Assign User To Groups 179
- View All User Permissions 181
- Maintain Groups 182
- View and Manage Locks 185
- Password Encryption 186 (for the third-party DBMS connection password; only available for Oracle and SQL Server Repositories for Enterprise Architect releases prior to 7.1)
- Create Workflow Scripts. 187

Other Security tasks can be performed by users who do not have Administrative rights. These tasks include:

- Lock Model Elements 192
- Lock Packages 193
- Apply a User Lock 194
- Identify Who Has Locked An Object 196
- Locked Element Indicators 195
- Manage Your Own Locks 196
- Change Your Own Password 190

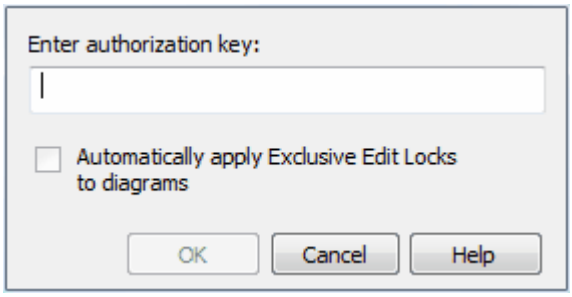
Notes:

- User security is not enabled by default in Enterprise Architect; you must enable it 173 first.
- For a number of operations in Enterprise Architect, if security is enabled a user must have the appropriate user or group access permission to perform the operation. However, if security is not enabled, the user does not have to have access permissions. See the *List of Available Permissions* 183 topic.

12.1 Enable Security

User security is not enabled by default in Enterprise Architect. To enable security for a project in Enterprise Architect for the first time, follow the steps below.

1. Access the *Registered Users* section of the Sparx Systems website (http://www.sparxsystems.com/registered/reg_ea_corp_ed.html), and obtain the Authorization Key. (You must have the Registered Users login and password to access this web site.)
2. In Enterprise Architect, select the **Project | Security | Enable Security** menu option. The Enter authorization dialog displays



3. In the **Enter authorization key** field, type the authorization key from the Sparx Systems website.
4. If required, select the **Automatically apply Exclusive Edit Locks to diagrams** checkbox.

Note:

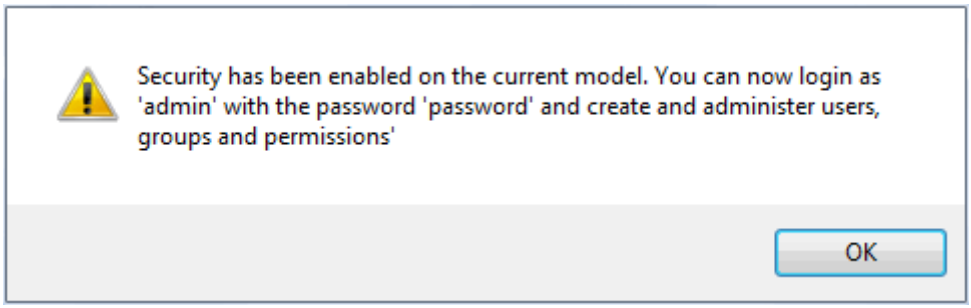
In standard (*User/Group Locking*) security mode 174, this option blocks multiple users from simultaneously attempting to modify the same diagram. As a user *modifies* a diagram, Enterprise Architect automatically applies a User Lock to the diagram, preventing any other user from modifying it. It is creating difference between the database and buffer versions of the diagram that triggers the temporary lock, and elimination of difference that releases the lock. Therefore, Enterprise Architect releases the lock when:

- The user saves the changes to the diagram, with the **Save** icon or keyboard keys
- The user undoes the last remaining action in the **Undo** list
- The user saves or discards changes via the system prompt when they close the diagram.

If the diagram already has a User Lock or Group Lock that does not exclude the current user, this lock is set aside and saved when the temporary User Lock is applied. When the temporary User Lock is released, the pre-existing lock is restored.

The option is ignored in *Require User Lock* security mode.

5. Click on the **OK** button. Security is enabled, and an Admin user and user group are created with full permissions (all access rights listed in List of Available Permissions 183) and a password of **password**.



6. Select the **Project | Security | Login as Another User** menu option, and log in as **Admin** with the initial password of **password**.

Note:

To change the Admin password, see the *Change Password* 190 topic.

7. Set up users and permissions as required.

Note:

Once security has been enabled, you must have the **Security - Enable/Disable** 183 access right to turn it off. The initial administrator automatically has this access right.

8. To disable security, click on the **Enable Security** menu option, and again type the authorization key in the Authorization dialog. Click on the **OK** button. Security is disabled.

Notes:

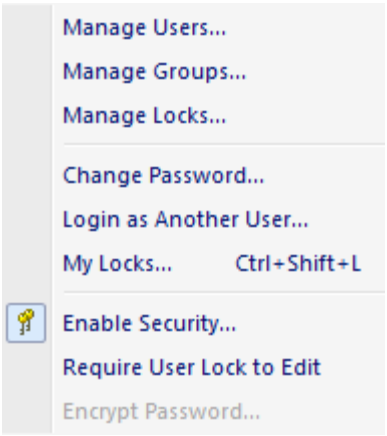
- The system prompts you to log off the project and log on again, but this is not strictly necessary.
- To re-enable security, follow the procedure above, but be aware that any changes you have made to the admin user (password and reduced access rights) are reset to **password** and full access.
- The **Automatically apply Exclusive Edit Locks to diagrams** option is not displayed when disabling security. Therefore, to toggle the setting whilst security is enabled you must disable security and re-enable it. Security settings (users, groups and permissions) and locks on elements, are NOT affected by this action.

12.2 Security Policy

There are two possible security policies in Enterprise Architect:

1. In the *User/Group Locking* mode, all elements and diagrams are considered unlocked and anyone can edit any part of the project. However, when you edit a diagram, package or element, you lock the element or set of elements at either the user level or group level. This mode is good for cooperative work groups where there is a solid understanding of who is working on which part of the model, and locking is used mainly to prevent further changes or to limit who has access to a part of the model.
2. The *Require User Lock* mode is more rigorous. The Enterprise Architect model is read-only - everything is locked so that nobody can edit anything unless they explicitly check out the object with a user lock. A single 'check out' function operates on a diagram to check out the diagram and all contained elements in one go. There are also functions on the context (right-click) menus of packages, diagrams and elements in the Project Browser to apply a user lock when this mode is in use. You would use this mode when there is a strict requirement to ensure only one person can edit a resource at one time. This is suitable for much larger projects where there might be less communication between users.

Toggle between these modes using the **Project | Security | Require User Lock to Edit** menu option - deselected for User/Group Locking mode, and selected for Require User Lock mode.



Notes:

- When you add new elements in Mode 1 (**Require User Lock to Edit** deselected, elements editable by default), no user lock is created automatically for the newly created element.
- When you add new elements in Mode 2 (**Require User Lock to Edit** selected, elements locked by default), a user lock is created on the new element to enable instant editing.

12.3 Maintain Users

If you enable security you have access to the Security Users dialog, which you can use to set up more users for your model.

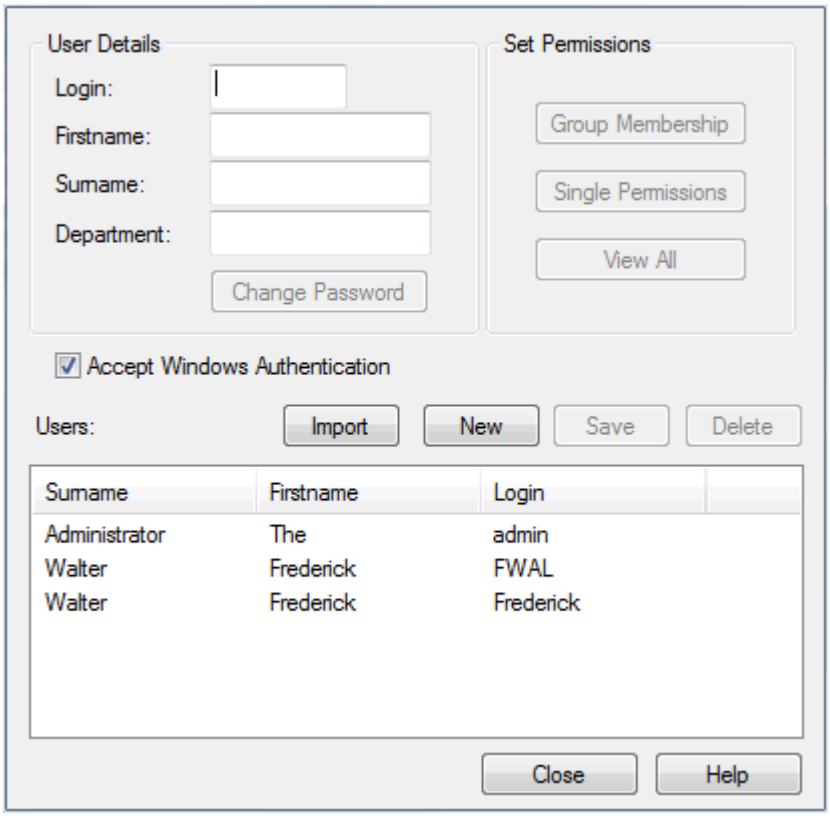
Note:

You must have **Security - Manage Users** 183 permission to maintain users, and **Change Password** 183 permission to change the password of the current user; the initial **Admin** administrator automatically has these permissions.

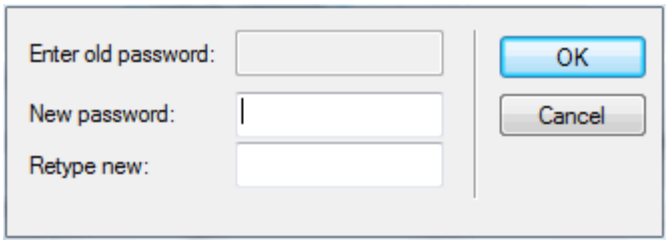
Set Up a User

To set up a user for your model, follow the steps below:

1. Select the **Project | Security | Manage Users** menu option. The Security Users dialog displays.



2. You can use the Security Users dialog to set up new users by providing their name and other details. You can also import user IDs from a Windows Active Directory 177, assign User IDs to groups 179, set up **Single Permissions** 180 or **View All** 181 permissions for the currently selected user.
3. To identify a new user on this dialog, click on the **New** button and type in the user's login ID, first name and last name. If required, also provide the user's department name.
4. To set the user's password, click on the **Change Password** button. The Change Password dialog displays.



5. In the **New password** field, type the user's password. This must be 12 characters or less in length.
6. In the **Retype new** field, type the user's password again, for confirmation.
7. Click on the **OK** button.
8. A *'Password Changed'* message displays. Click on the **OK** button.
9. When you have entered the details for the user, click on the **Save** button. Either click on the **New** button to add another user, or the **Close** button to exit the Security Users dialog.

Notes:

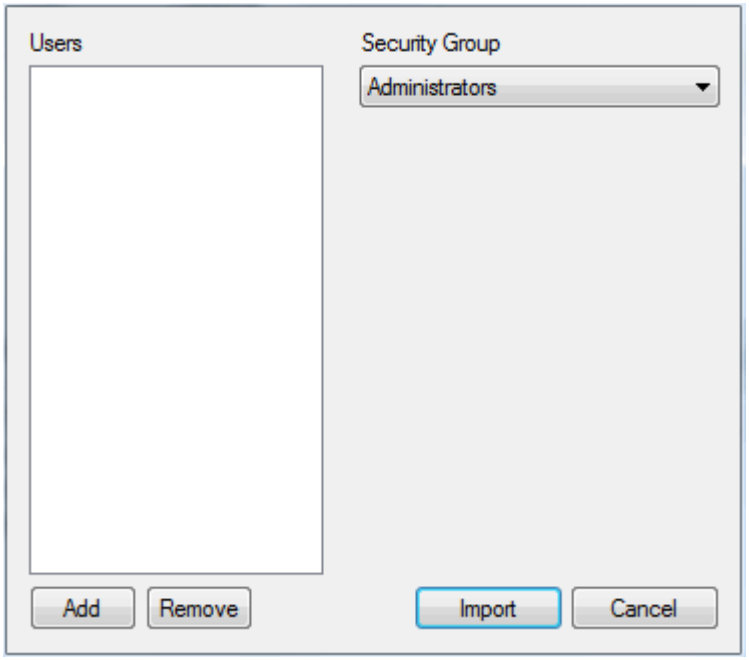
- You can transport the user definitions between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.
- If you select the **Accept Windows Authentication** checkbox, when a user opens the model Enterprise Architect checks the users database for their Windows ID and, if it matches, automatically logs the user in without prompting for a password.
- The **Accept Windows Authentication** checkbox enables the **Import** button, which you can select to import user IDs from a Windows Active Directory.
- As a security measure, the **Accept Windows Authentication** checkbox is automatically deselected if the project .eap file is moved to a different location. Once the file has been relocated, you can select the checkbox again to apply Windows authentication from the new database.

12.4 Import User IDs From Active Directory

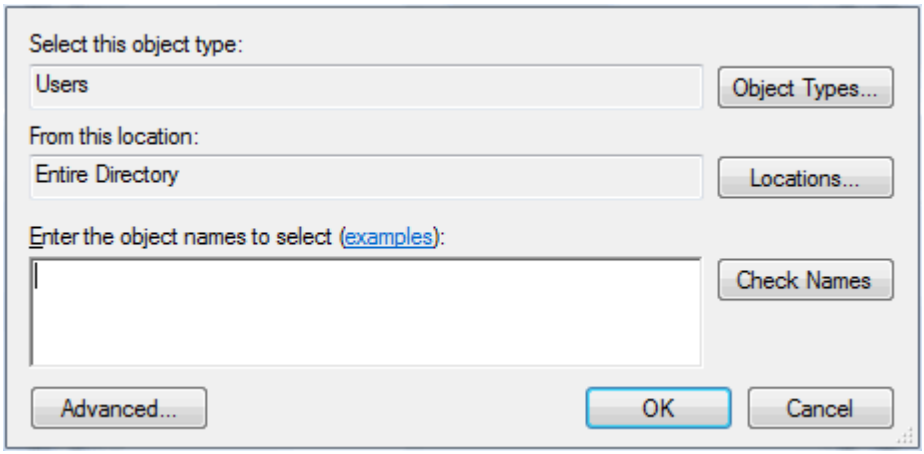
When you import user IDs from Windows Active Directory, you should create an appropriate user group 182 and assign the imported user IDs to that group. You can then assign appropriate permissions to the group. When a user logs in to Enterprise Architect under their Windows login ID, they do not have to enter a password; Enterprise Architect automatically generates a random password. However, you can assign a new password to an imported user ID if required.

To import user IDs from a Windows Active Directory, follow the steps below:

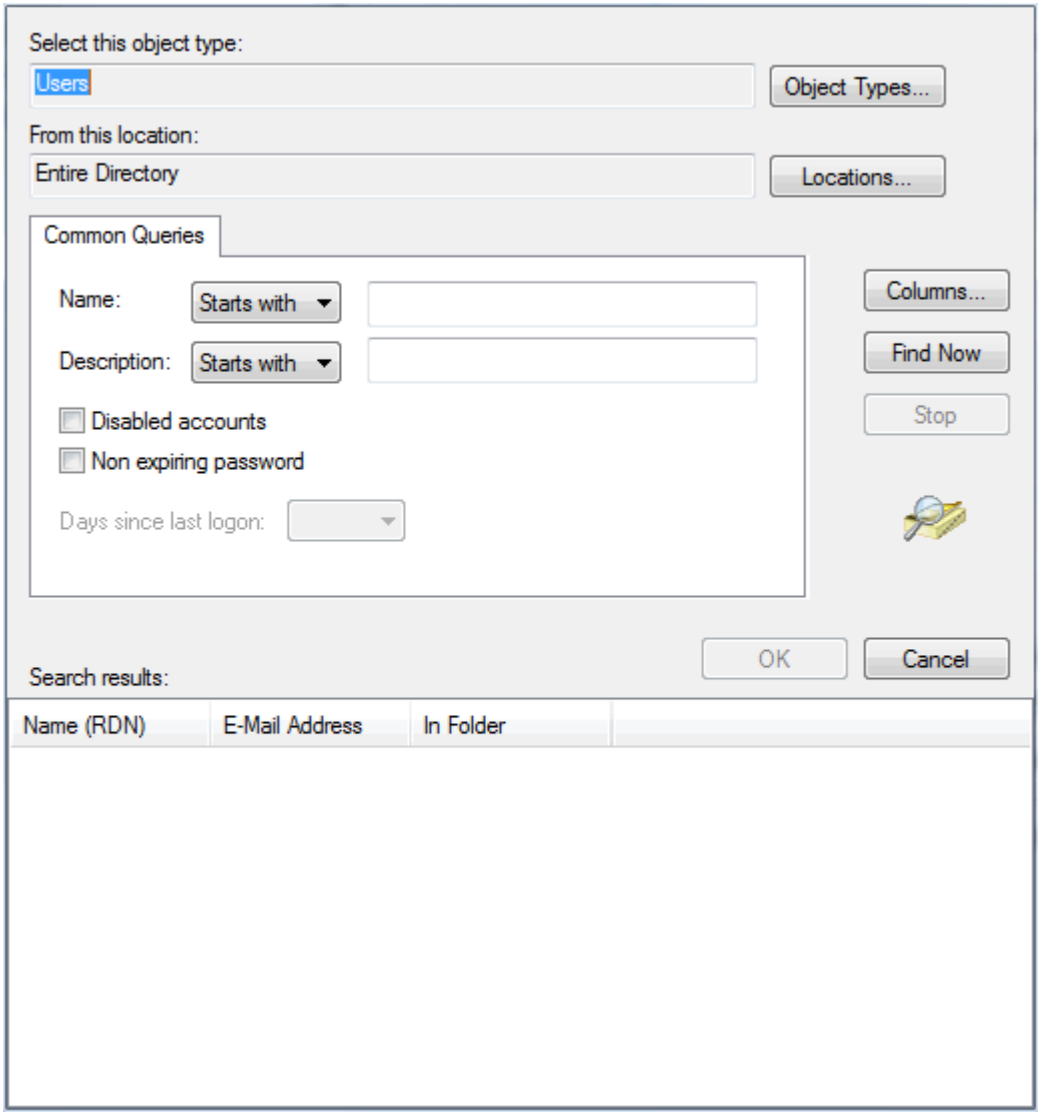
1. On the Security Users dialog select the **Accept Windows Authentication** checkbox and click on the **Import** button. The Import Users dialog displays.



2. On the Import Users dialog, click on the down arrow in the **Security Group** field and select the appropriate security group for the imported user IDs.
3. Click on the **Add** button. The Select Users screen displays.



4. Click on the **Object Types** button, and on the Object Types dialog select the checkbox for the type of object to import from the Active Directory. Click on the **OK** button to return to the Select Users dialog.
5. Click on the **Locations** button, and on the Locations dialog browse for and select the checkbox for the location to import from within the Active Directory. Click on the **OK** button to return to the Select Users dialog.
6. In the **Enter the object names to select** field, either:
   - type in the user IDs individually (click on the **examples** link to see examples of the correct formats) or
   - click on the **Advance** button to search for IDs; the Select Users dialog redisplays with a Common Queries tab.



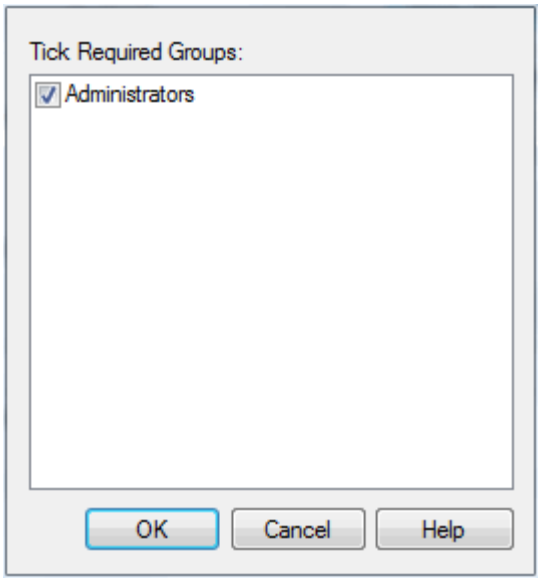
7. In the **Name** and **Description** fields, type any characters or text that helps identify the IDs you are searching for. Click on the drop-down arrow of the **Starts with** field and select the appropriate qualifier.
8. If required, select the **Disabled accounts** or **Non-expiring password** checkboxes, and/or select a value in the **Days since last logon** field, to further filter the IDs to search for.
9. Click on the **Find Now** button to initiate the search, and to display a list of IDs in the bottom panel of the dialog. You can vary the types of information shown here by clicking on the **Columns** button and selecting the column headings to display.
10. When you have identified the IDs to import, click on a required ID (or press **[Ctrl]** or **[Shift]** while you click to select several) and click on the **OK** button. The Select Users dialog redisplays, with the selected ID or IDs listed in the **Enter the object names to select** field.
11. Click on the **OK** button to redisplay the Import Users dialog with the selected users' names listed in the Users panel.
12. Click on the **Import** button to add the user IDs to the Security Users dialog. Click on a user ID to populate the dialog fields with the user ID details, and set group permissions 182 as required.

12.5 Assign User To Groups

To set up user groups follow the steps below:

1. Select the **Project | Security | Manage Users** menu option. The Security Users dialog displays.
2. Click on the **Group Membership** button. The User Groups dialog displays.

3. Select the checkbox against each group this user belongs to.



4. Click on the **OK** button to assign the user to each group.

Notes:

- To create new user groups, see the *Maintain Groups* 182 topic.
- You can transport these user groups between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.

12.6 Set Up Single Permissions

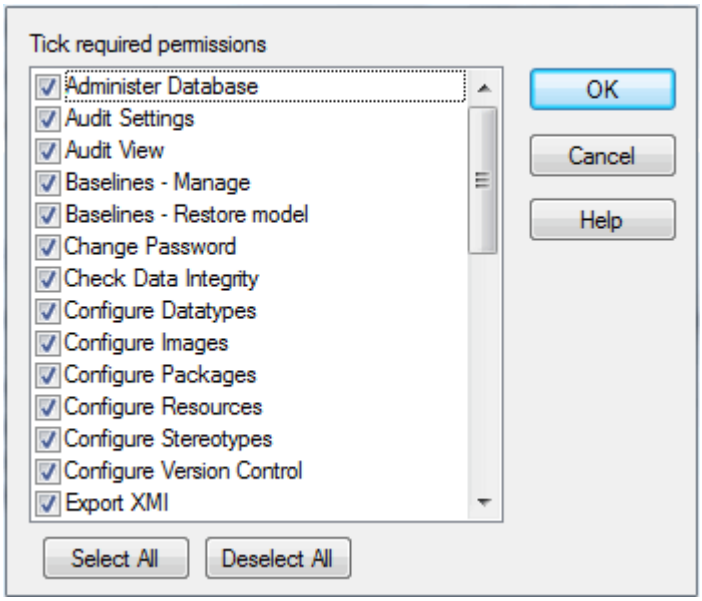
You can set specific user permissions from the User Permissions dialog. Specific user permissions are added to permissions from group membership to provide an overall permission set.

Note:

You must have **Security - Manage Users** 183 permission to assign permissions to users; the initial **Admin** administrator automatically has this permission.

To set up single permissions for a user follow the steps below:

1. Select the **Project | Security | Manage Users** menu option. The Security Users dialog displays.
2. Click on the **Single Permissions** button. The User Permissions dialog displays.



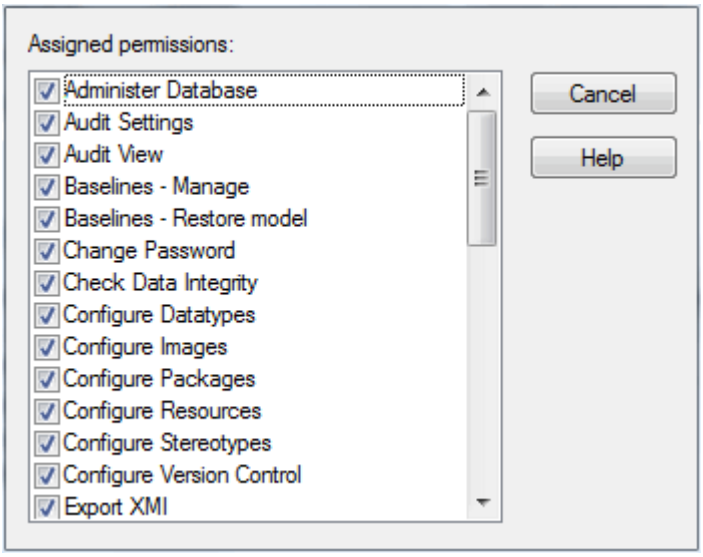
3. Select the checkbox against each specific permission to apply to this user. Click on the **Select All** button to select all permissions for the user, or click on the **Deselect All** button to clear all selected permissions.
4. Click on the **OK** button to assign the selected permissions to the user.

Notes:

- A user's total permissions are those granted by Group Membership plus those granted by specific permission assignment.
- You can transport these user permissions between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.

12.7 View All User Permissions

The All user permissions dialog shows a list of all permissions a user has, derived from their individual profile and from their membership of security groups. To display the dialog, select the **Project | Security | Manage Users** menu option, then select the required user and click on the **View All** button.



12.8 Maintain Groups

Security groups make it easy to configure sets of permissions and apply them to a number of users in one action.

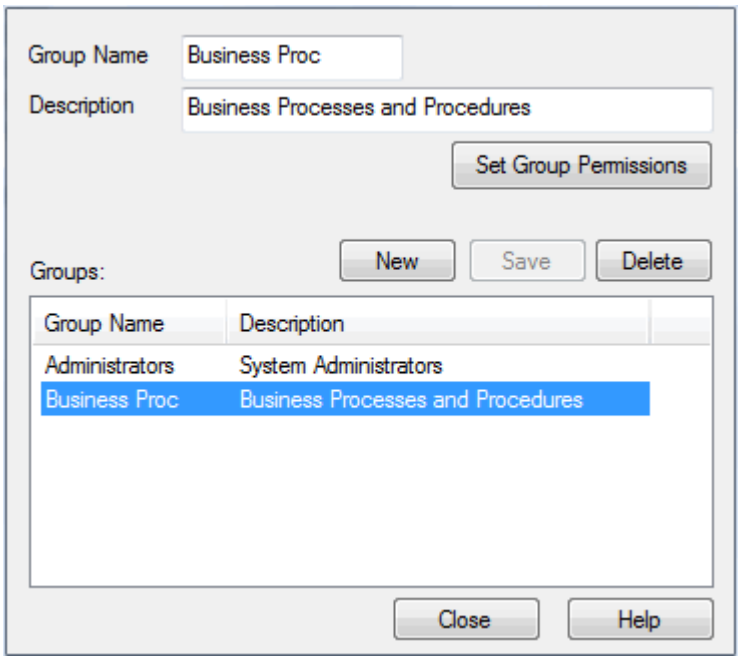
Notes:

- You must have **Security - Manage Users** 183 permission to manage user groups; the initial **Admin** administrator automatically has this permission.
- You do not define groups as group logins with passwords. If you intend to use a group login, you can define a single-user login and password 175 that all group members use (that is, Enterprise Architect allows multiple logins under one user ID).

Set Up a Security Group

To set up a security group, follow the steps below:

1. Select the **Project | Security | Manage Groups** menu option. The Security Groups dialog displays.



2. In the **Group Name** and **Description** fields, type the security group name and a description of the group.
3. Click on the **Save** button.

Note:

You can transport these security group definitions between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.

12.9 Set Group Permissions

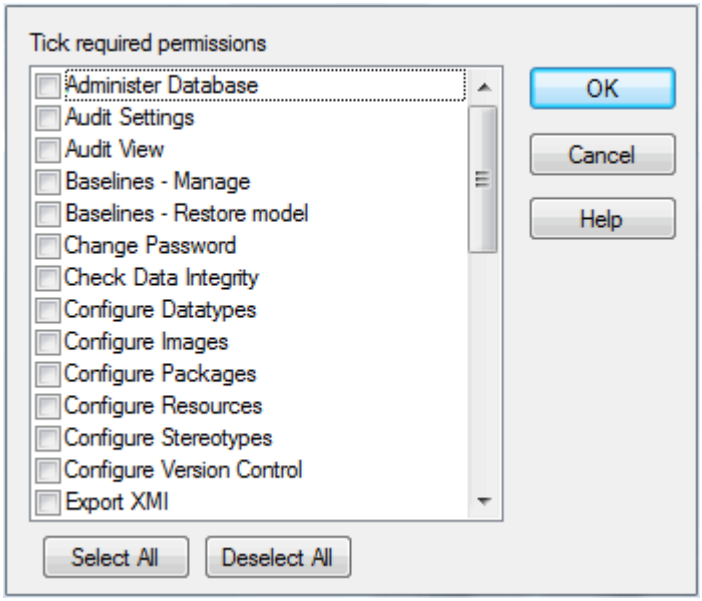
Note:

You must have **Security - Manage Users** 183 permission to assign permissions to user groups; the initial **Admin** administrator automatically has this permission.

To set up permissions to apply to a security group, follow the steps below:

1. Select the **Project | Security | Manage Groups** menu option. The Security Groups dialog displays.

2. Click on the **Set Group Permissions** button. The Group Permissions dialog displays.



3. Select the checkbox against each required permission. Click on the **Select All** button to select all permissions for the user, or click on the **Deselect All** button to clear all selected permissions.
4. Click on the **OK** button to assign the permissions. All of the users assigned to this group share in this set of permissions.

Note:

You can transport these group permission definitions between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.

12.10 List of Available Permissions

The following table lists the available permissions in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect. These permissions are required for the corresponding operations if security is enabled.

Note:

Some permissions take precedence over others. For example, if you set **Use Version Control** permission for a user, that user can modify model elements on import even if they do not have **Update Element** permission.

Permission	Enables the user to
Administer Database	Compact 84 and repair 85 project database.
Admin Workflow	Develop and manage workflow scripts 187.
Audit Settings	Change the audit settings in the Audit Settings 198 dialog.
Audit View	Enable auditing and display data in the Audit View 203 and Audit History 205 tab.
Baselines - Manage	Create, delete, import and export Baselines 207.
Baselines - Restore	Merge data 207 into the project model from a Baseline or XML file.
Change Password	Change your own password 175 or (Administrator) another user's password.

Permission	Enables the user to
Check Data Integrity	Check and repair project integrity 76.
Configure Datatypes	Add, modify and delete datatypes 249.
Configure Images	Configure alternative element images.
Configure Packages	Configure controlled packages 112 and package properties.
Configure Resources	Create and manage Resources window items: RTF templates, patterns, profiles, favorites.
Configure Stereotypes	Add, modify and delete Stereotypes 245.
Configure Version Control	Set up version control options 135 for the current model.
Export XMI	Export 105 model to XMI.
Generate Documents	Generate RTF and HTML documents from model packages.
Generate Source Code and DDL	Generate source code and DDL from model element. Synchronize code against model elements if it already exists.
Import XMI	Import 106 model from XMI.
Lock Objects	Lock an element 192 or package 193.
Manage Diagrams	Create new diagrams, copy existing and delete diagrams. Also save diagram as UML Pattern.
Manage Issues	Update and delete Issues.
Manage Project Information	Update and manage resources, metrics, risks.
Manage Reference Data - Update	Update and delete reference items 226.
Manage Replicas	Create 101 and synchronize 102 replicas.
Manage Tests	Update and delete Test records.
Reverse Engineer from DDL and Source Code	Reverse engineer from source code or ODBC, and synchronize model elements against code.
Security - Enable/Disable	Disable 173 user security in Enterprise Architect.
Security - Manage Locks	View and delete 185 element locks.
Security - Manage Users	Maintain users 175, groups 179 and assigned permissions 180.
Spell Check	Spell check package and set spell check language.
Transfer Data	Transfer model 79 between different repositories.
Transform Package	Perform transformations of packages and elements.
Update Diagrams	Update diagram appearance, properties and layout, including the Page Setup dialog.
Update Element	Save model changes (including delete) for elements, packages, and relationships.
Use Version Control	Check files in and out 162 using version control.

12.11 View and Manage Locks

From time to time it might be necessary to examine or delete locks placed on elements by users. Enterprise Architect provides a function to view and manage active locks.

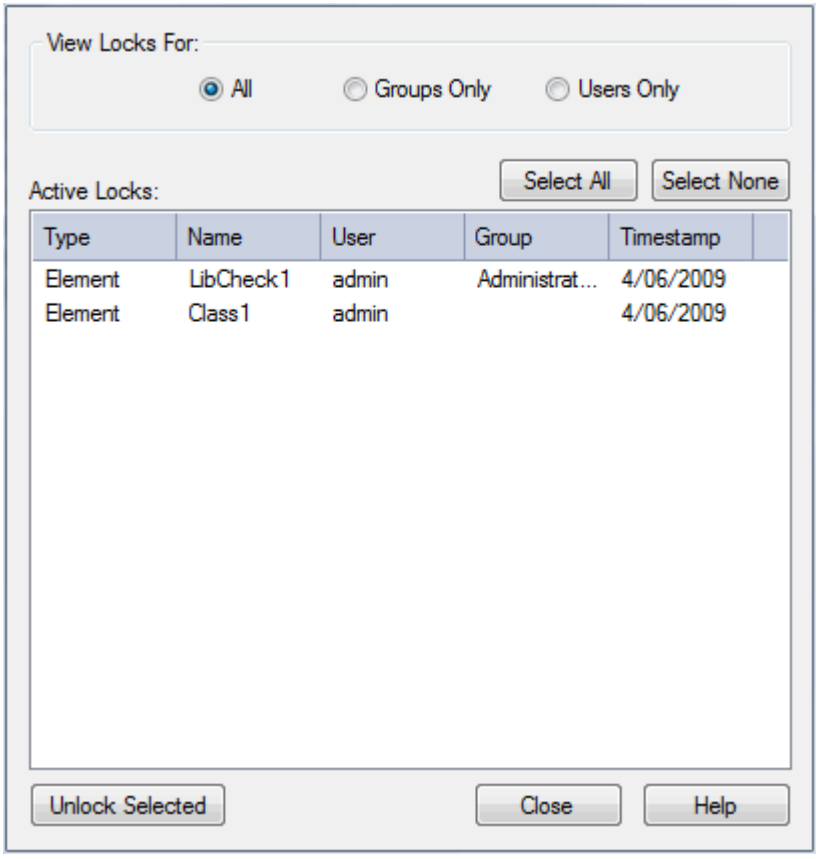
Notes:

- You must have **Security - Manage Locks** 183 permission to view and delete user locks; the initial **Admin** administrator automatically has this permission.
- If an element is locked, connectors attached to it are also locked. To unlock the connector, you must unlock the element. However, under certain circumstances you can add new connectors to a locked element 193.

Delete a Lock

To view locks and, if necessary, delete them, follow the steps below:

1. Select the **Project | Security | Manage Locks** menu option. The Active Locks dialog displays.



2. In the View Locks For panel, click on the radio button for the type of lock to view: **All**, **Groups Only** or **Users Only**. Locks of the appropriate type are listed in the Active Locks panel. If you want to display the resulting information in a more readable layout, you can resize the dialog and its columns.
3. To remove a lock, click on it and click on the **Unlock Selected** button.
4. When finished, click on the **Close** button to close the dialog.

12.12 Password Encryption

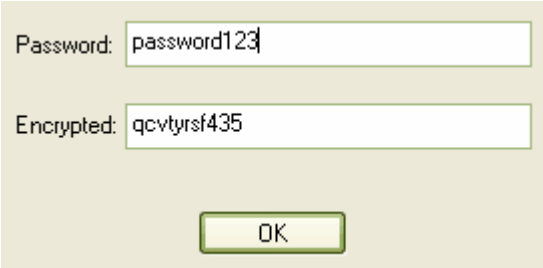
Note:

This topic is retained to support regression to releases of Enterprise Architect prior to version 7.1. For password encryption for all repositories at and beyond release 7.1, see the *Save Model Copy or Shortcut* topic.

Users of SQL Server or Oracle repositories have the option of encrypting the password used to set up the connection between Enterprise Architect and the repository. The Enterprise Architect user does not have the real password, thereby preventing them from accessing the repository using other tools such as Query Analyzer or SQLPlus.

Once security is enabled, the administrator must log on to access the dialog to create encrypted passwords. To encrypt a password, follow the steps below:

1. Select the **Project | Security | Encrypt Password** menu option. The following dialog displays:

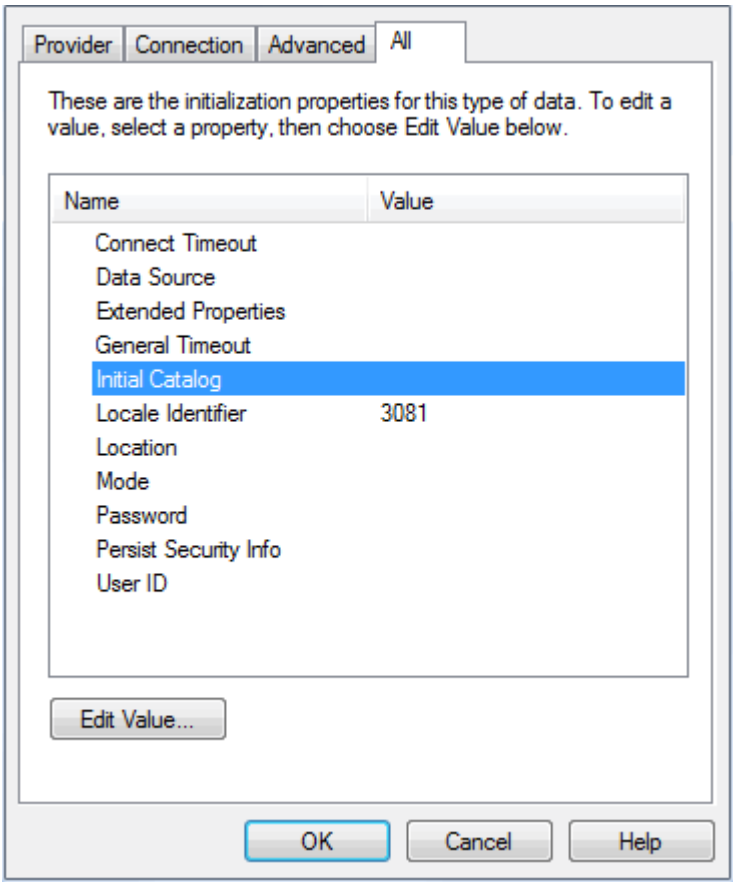


2. In the example above, the password **password123** is used to access the repository.
3. To connect Enterprise Architect to the repository, the user enters the encrypted password prefixed with **$$**, so the encrypted password becomes **$$qcvtyrsf435**.

For more information relating to connecting to Oracle and SQL Server, see the *Connect to Oracle Data Repository* 55 and *Connect to SQL Server Data Repository* 52 topics respectively.

Notes:

- Do not use the **Test Connection** button as it can cause an error with encrypted passwords.
- For SQL Server repositories, you must enter the *Initial Catalog* details from the All tab of the Data Link Properties dialog.



12.13 Workflow Scripts - Introduction

Good corporate governance relies on well written and transparent project development guidelines and company policy. A project might be compromised if the appropriate policies and procedures are poorly understood and not followed correctly - effective governance can be hampered by human error and the costs of recovering from the inadequate compliance of developers.

Company policy and procedures can be integrated with the development process to manage work flows, determine access rights, extend role based security permissions and respond to property change events. This approach reduces compliance costs, enhances collaborative development and gives you confidence that projects are being developed correctly the first time around. Development teams can adhere to best practice guidelines that govern model validation, change management, access controls and general development principles.

Enterprise Architect enables you to create workflow scripts 188 that provide a more robust approach to applying company policy and strengthening project development guidelines by validating against the policy and procedures within the model itself. Project administrators can write scripts to manage the way users interact with a model, such as managing security, staff compliance and model access, and monitoring changes made by users. Administrators can also use workflow scripts to control a user's capacity to change a model element, taking into account factors such as access rights, group membership and even the value of a proposed change.

When a model is launched, the Workflow Engine is initialized with the current user and group memberships. This information determines who can access and modify parts of a given model. When a selected event occurs, the script engine is initialized with values including the author's name and access rights, and the element name and version details. The workflow script implements rules governing change management, access control and model validation. If a user attempts to make changes in violation of company policy, the script denies the update. The user is notified why the validation failed and the activity is logged. These reminders help to reinforce company policy, reduce human error and provide management with valuable project feedback.

12.13.1 Workflow Script Functions

Workflow scripts are executed by the Enterprise Architect workflow engine, to manage user input. You write the scripts in the Scripter window, in VBScript, under the Workflow group type.

Functions Enterprise Architect Calls to Validate and Control User Input

For each of the following functions that Enterprise Architect calls, a set of objects 188 are filled.

Function	Use to...	Return Value
AllowPhaseUpdate(OldValue, NewValue)	Validate a change a user has made to a phase.	• **True** to allow this user to make this change. • **False** to disallow the change and revert to the previous value.
AllowStatusUpdate(OldValue, NewValue)	Validate a change a user has made to a status.	
AllowTagUpdate(TagName, OldValue, NewValue)	Validate a change a user has made to a Tagged Value.	
AllowVersionUpdate(OldValue, NewValue)	Validate a change a user has made to a version.	
CanEditPhase()	Enable or disable the control for editing a phase.	• **True** to allow this user to make changes by enabling the control. • **False** to completely disable edit of this property by disabling the control.
CanEditStatus()	Enable or disable the control for editing a status.	
CanEditTag(TagName)	Enable or disable the control for editing a Tagged Value.	
CanEditVersion()	Enable or disable the control for editing a version.	
PreAllowPhaseUpdate(OldValue, NewValue)	Determine what information is required to validate this change.	Semi-colon separated list of additional data required in order to validate this change. See the list of supported data types 188.
PreAllowStatusUpdate(OldValue, NewValue)		
PreAllowTagUpdate(TagName, OldValue, NewValue)		
PreAllowVersionUpdate(OldValue, NewValue)		

Functions Enterprise Architect Calls to Create a Search With User Tasks

Function	Use to...	Return Value
GetWorkflowTasks	Describe the searches that this user must run.	Ignored

Supported Data Types

Tests - fill the *Tests* array in the *WorkflowContext* object.

Workflow Data Structures - Objects Enterprise Architect Fills

WorkflowUser

This object provides information about the user currently logged in to the model. It is filled by Enterprise Architect before any function is called by Enterprise Architect. It has the following properties:

- **Username** - the username for login to the system (if using Windows Authentication, this matches the Windows username)
- **Firstname** - as found in the Security Users 175 dialog
- **Surname** - as found in the Security Users dialog
- **Fullname** - the combination **<Firstname> <Surname>** (the form Enterprise Architect uses for **Author** fields and similar).

This object also calls the following function:

Function	Use to...	Return Value
IsMemberOf(GroupName)	Check group membership of the current user.	**True** if the current user is a member of the group with the specified name.

WorkflowContext

This object provides information about the object currently in context. It is filled by Enterprise Architect before any searches except **GetWorkflowTasks** 188 are run. It has the following properties:

- **MetaType** - the type of the current object, either an Enterprise Architect core type or a profile-specified metatype
- **Name** - as found in the object Properties dialog
- **Status** - as found in the object Properties dialog
- **Phase** - as found in the object Properties dialog
- **Version** - as found in the object Properties dialog
- **GUID** - the GUID of the object
- **Stereotypes** - an array of strings for the stereotypes applied to this object
- **Tags** - an array of Tagged Values, providing:
  - **Name** - the Tagged Value name
  - **Value** - the Tagged Value value
- **Tests** - an array of tests; only filled during an *Allow** call after the *PreAllow** call has specified that tests are required. Provides the following details, as found in the Testing window:
  - **Name**
  - **Status**
  - **RunBy**
  - **CheckedBy**
  - **TestClass**
  - **TestType**

The WorkflowContext object also calls the following function:

Function	Use to...	Return Value
TagValue(TagName)	Get the value from a named tag.	Returns the value of the first Tagged Value with that name, or an empty string if no Tagged Value with that name exists.

Workflow Data Structures - Objects You Can Fill

WorkflowStatus

Use this to provide information on the status of the object.

- **LogEntry** - set to **True** or **False**, to indicate whether a log item should be recorded
- **Reason** - indicate what reason should be recorded in the log
- **Action** - indicate how to display the log message; valid values are: **MessageBox**, **StatusBar**, **Output** (default).

WorkflowSearches

Provides an array of searches. Use **Redim WorkflowSearches(x)** to specify the number of searches being provided. Each search has the following attributes:

- **Name** - the name of this search
- **Group** - the name of the group that this search should appear under in the **Search** combo box
- **ID** - the unique GUID for this search
- **Tasks** - the array of tasks that this search looks for; an entry describes how to find all objects required to meet a particular task:
  - **Name** - the name of the task, as displayed in the Search view; workflow searches are grouped by this field by default
  - **Conditions** - an array of conditions, all of which must be matched for an object to be included in this task; a condition is a comparison of a single field to a value:
    - **Column** - the name of the field
    - **Operator** - operator types, either **=** (provide matching values only) or **<>** (provide non-matching values only)
    - **Value** - if this contains a comma, the string is treated as a comma separated list of values to compare against; otherwise the string is a single value to compare against.

Functions Enterprise Architect Provides For You to Call

Enterprise Architect provides the subfunction **SetLastError(message, outputMethod)** for you to call, to log and/or report the provided message to the user.

You can also call the following functions:

Function	Use to...	Return Value
NewSearch(name, group, guid, taskcount)	Create a new search object to be included in WorkflowSearches. Initializes each member.	The created search.
NewTask(name, conditioncount)	Create a new task object to be included in a search. Initializes each member.	The created task.
NewCondition(column, operator, value)	Create a new condition object to be included in a task. Initializes each member.	The created condition.

12.14 Change Password

There are two ways in which a user's password can be changed, when security is set:

- A user can select the **Change Password** menu option and change their own password
- The Administrator can set or change any user's password, on the Maintain Users dialog.

Note:

A user must have **Change Password** 183 permission to change a password; the initial **Admin** administrator automatically has this permission.

User Change

If security is set and you want to change your own password, follow the steps below:

1. Select the **Project | Security | Change Password** menu option. The Change Password dialog displays.

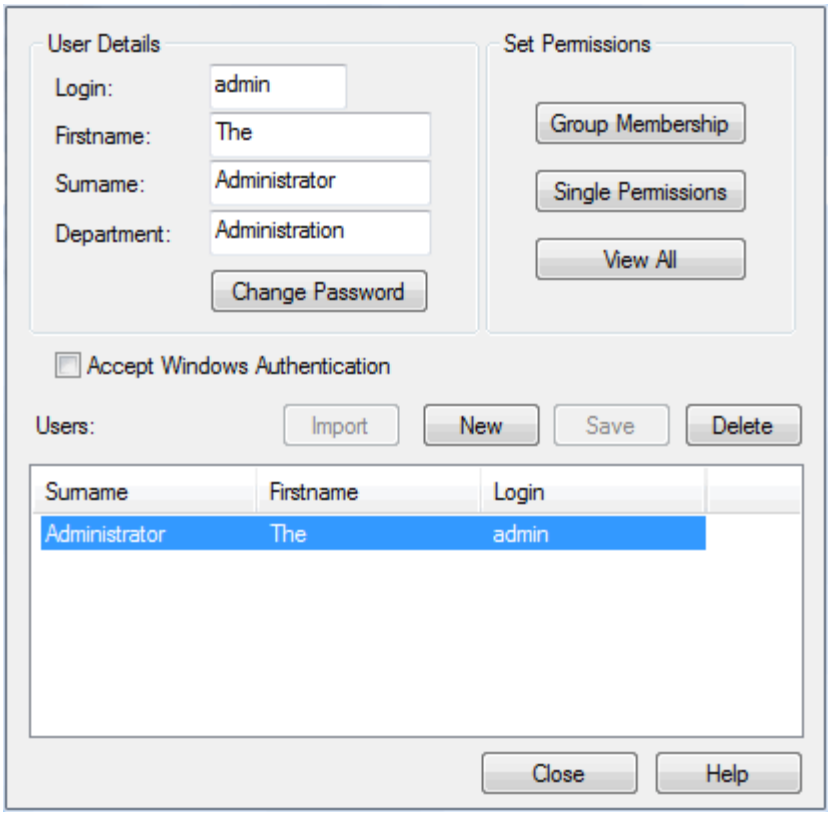
2. In the **Enter old password** field, type your current password.
3. In the **New password** field, type your new password (this must be 12 characters or less in length).
4. In the **Retype new** field, type your new password again, for confirmation.
5. Click on the **OK** button.
6. A *'Password Changed'* message displays. Click on the **OK** button to clear the message.

Your new password is effective next time you log in.

Administrator Change

To set or change any user's password, follow the steps below:

1. Select the **Project | Security | Manage Users** menu option. The Security Users dialog displays.



2. Click on the user name in the Users: panel, to display the user details in the dialog fields.
3. Click on the **Change Password** button. The Change Password dialog displays.

4. In the **New password** field, type the user's password; this must be 12 characters or less in length.

> **Note:**
> You do not have to enter the user's current password, as they might have forgotten it and therefore it is possible that nobody can provide that value.

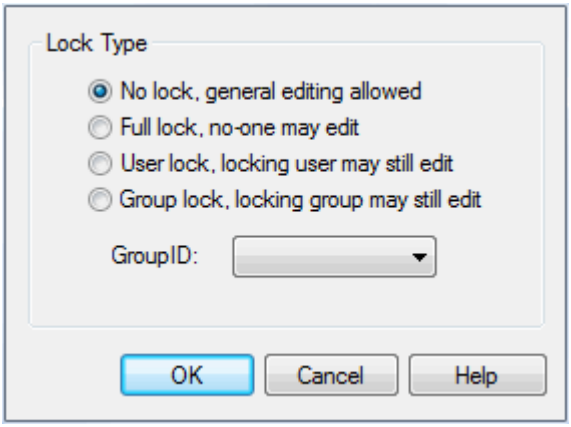
5. In the **Retype new** field, type the user's password again, for confirmation.
6. Click on the **OK** button.
7. A *'Password Changed'* message displays. Click on the **OK** button.

12.15 Lock Model Elements

> **Note:**
> When security is enabled, you must have **Lock Objects** 183 permission to lock an element.

You can lock a package, element or diagram using the corresponding **Lock** context menu option in the Project Browser, and you can lock an element or diagram using the corresponding **Lock** context menu option in the diagram.

Under the standard security policy (**Require User Lock to Edit** deselected), when you select the **Lock** option the Element Lock dialog displays:



The four lock options available are:

- **No lock** - do not lock this element; clear any existing lock
- **Full lock** - lock this element so that no-one can edit it
- **User lock** - lock this element so that only the locking user can make further edits
- **Group lock** - lock this element so that any member of the specified group (in the **GroupID** field) can update the element, but others are excluded.

Select the appropriate lock and click on the **OK** button.

If the item is already locked, only the appropriate lock option and **No lock** are available. You have to release the lock in order to set a different type of lock.

Under the rigorous security policy, a different dialog displays. See the *Apply a User Lock* 194 topic.

If a diagram is locked and you select an object on it, the object border displays in red. This indicates that you cannot change the object.

12.16 Add Connectors To Locked Elements

When working with locked elements, the ability to add connectors depends on the locked status of the source and target elements. The rules are:

- **Source unlocked, target unlocked**: any kind of connector can be added
- **Source unlocked, target locked**: allowed, except for composition connectors
- **Source locked, target unlocked**: prohibited, except for composition connectors
- **Source locked, target locked**: prohibited for all connectors.

That is, a connector can be added if its source is unlocked, regardless of the locking state of the destination (think of it as modifying what the source can see). The exception is composition connectors, where the target (that is, parent) must be unlocked (think of it as modifying the parent by adding children).

Connectors with locked source or target elements are also locked. To unlock the connector, you must unlock 185 the source and/or target element.

12.17 Lock Packages

Note:

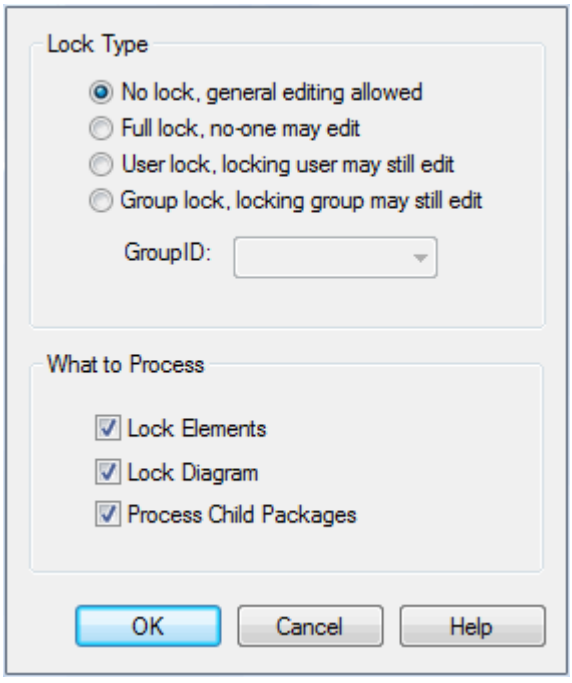
If security is enabled you must have **Lock Objects** 183 permission to lock a package.

You can lock all the contents of a package (and optionally all contents in child packages) in one step, using the *Lock Package* function. The locks are automatically applied to elements and to diagrams, as if they had been individually set or cleared. Lock types and details are the same as for locking a single element 192.

Lock a Package

To lock a package, follow the steps below:

1. Deselect the **Project | Security | Require User Lock to Edit** menu option.
2. In the Project Browser, right-click on the package to lock. The context menu displays.
3. Select the **Lock Package** menu option. The Lock/Unlock Package(s) dialog displays.

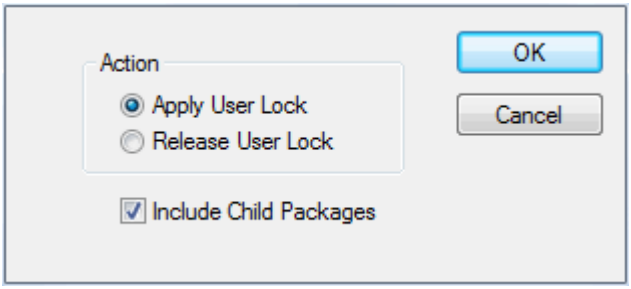


4. In the Lock Type panel, select the appropriate radio button for the lock to apply.
5. As required, select the checkboxes to lock elements and/or diagrams, and to process child packages (that is, lock the whole branch).
6. Click on the **OK** button to apply the lock.

12.18 Apply a User Lock

In the *Require User Lock to Edit* 174 security mode, where a User Lock is required before any edit can occur, you can set or release the lock in either a diagram or the Project Browser. Enterprise Architect adjusts the lock for the element, or for the diagram and any elements contained in the diagram.

In a diagram, you right-click on the element or diagram; in the Project Browser, you right-click on the package, diagram or element. In each case, select the **Apply/Release User Lock** context menu option for the selected item. The following dialog displays.



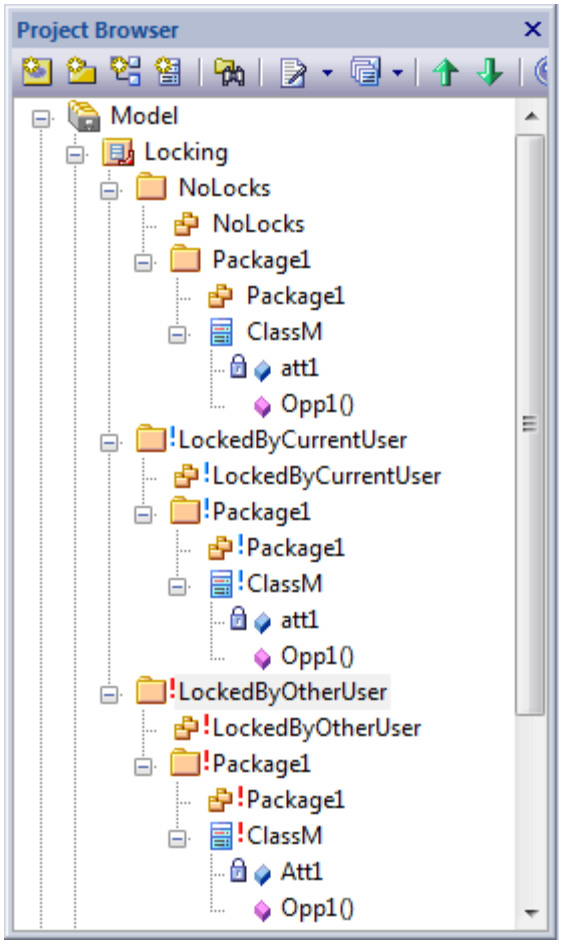
Select the appropriate radio button to apply or release a user lock on the selected item.

Note:

For a package, you can elect to also lock all child packages at the same time. If any elements in the package tree are locked by other users, a list of elements that couldn't be locked displays at the end of the process.

12.19 Locked Element Indicators

When an item is locked through Project Security, the lock is indicated in the Project Browser by a marker against the item, as shown below.



The meaning of the marker depends on the security mode.

If you are using the *Require User Lock to Edit* 174 security mode:

- **No** marker - there is no lock, the item is *not* editable, but any user can now apply a user lock 194 to edit the item
- **Blue** exclamation mark - the current user has applied a user lock and can edit the item; no other user can edit the item
- **Red** exclamation mark - another user has applied a user lock, and the current user cannot edit the item.

If you are using the *standard* 174 security mode 174:

- **No** marker - there is no lock, the item *is* editable, but any user can now apply a user or group lock 194
- **Blue** exclamation mark - the item has a lock set by the current user or a group having the current user as a member, and the user can edit the item
- **Red** exclamation mark - the item has a lock set by another user, or a group of which the current user is not a member; the current user cannot edit the item.

If another user has locked an item, you can identify who has locked it 196.

Note:

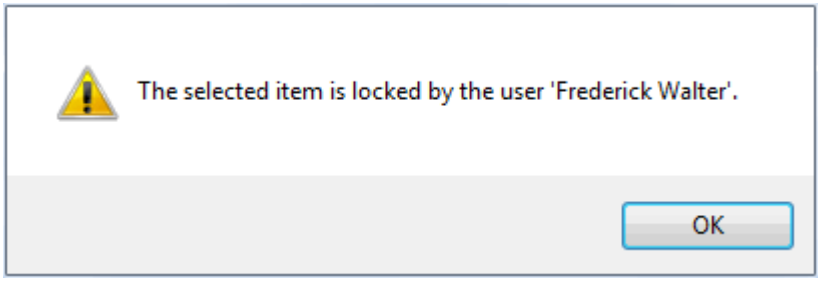
If a diagram is locked and you select an object on it, the object border displays in red. This indicates that you cannot change the object.

12.20 Identify Who Has Locked An Object

If you find that a diagram, package or element is locked, you can find out which group or user currently holds the lock on that item. To do this, follow the steps below:

1. In the Project Browser, right-click on the diagram, package or element that is locked by another user or user group. The context menu displays.
2. Select the **Lock** menu option.

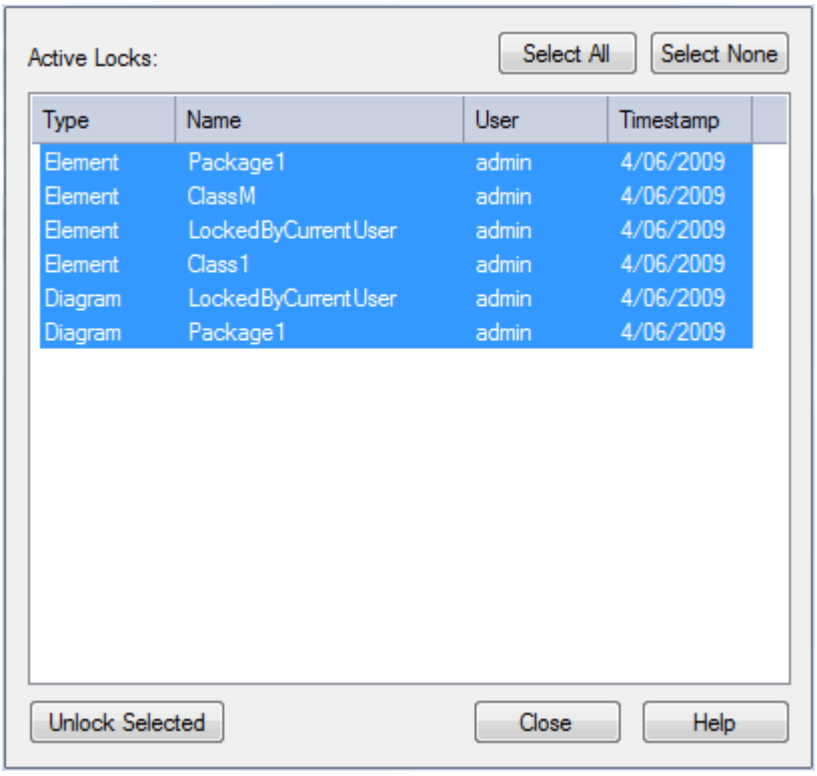
   A message box displays showing which group or user currently holds the lock on that item.



12.21 Manage Your Own Locks

You can view and delete your own user-level locks in Enterprise Architect. This is especially useful when working in Mode 2 security 174 (user locks required to edit).

To manage your locks select the **Project | Security | My Locks** menu option. The My Locks dialog displays.



In the My Locks dialog you can select one or more locks and delete them (that is, unlock the object) by clicking on the **Unlock Selected** button.

13 Auditing

Auditing is a project-level feature, available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions, that enables model administrators to record model changes in Enterprise Architect. By enabling this option, you can view information on changes such as:

- *Who* changed an element
- *How many* elements they changed
- *When* they changed the data
- *What* the previous values were, and
- *What type* of elements they changed.

Auditing does not record changes to RTF Templates, Model documents, Baselines or Profiles.

Warning:

If your site runs separate editions of Enterprise Architect, when Auditing is turned on in a project any Desktop or Professional edition users are locked out of the project. To restore access, turn Auditing off in the project from a Corporate, Business and Software Engineering, Systems Engineering or Ultimate edition instance of Enterprise Architect.

Auditing Quickstart

To quickly enable auditing and see it in action, see Auditing Quickstart 198.

Audit Settings

Once auditing is enabled within a project, you have a variety of options available for customizing what is recorded by the audit. See Audit Settings 198 for more information on the different settings available.

The Audit View

To view what has been recorded by the audit, use the Audit View 201, which provides an interface to everything recorded by auditing.

Note:

If security is enabled, you must have **Audit View** 183 permission to display data in the Audit View.

You can also obtain a snapshot of selected items in the model, using the Model View facility. In the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions of Enterprise Architect, this facility enables you to automatically generate this snapshot at intervals and, if there are changes in the items collected by the defined search, to trigger a notification to you of such changes. This enables you to monitor work flow 216 and other events of concern to you.

RTF Report

You can generate an RTF report that includes the audit history information for the selected element or package, by choosing the *basic + audit* RTF template.

Audit History

Using Auditing, you can track changes to an element or connector over time. However, enabling Auditing also enables an Audit History 205 tab in the Output window, which summarizes all changes made to the selected element or connector.

Performance Issues

By enabling auditing on a project, you increase the time taken for most actions. For most modeling tasks, this increase is insignificant. However, there are some instances where the difference is more substantial. See Performance Issues 206 for more information.

13.1 Auditing Quickstart

To quickly enable auditing and see it in action, follow the instructions below.

Enable Auditing

To enable Auditing:

1. Select the **View | Other Project Tools | Audit View** menu option to open the Audit View 201.
2. Click on the **Audit Settings** button. The Audit Settings 198 dialog displays.
3. Select the **Enable Auditing** checkbox.
4. Click on the **OK** button to close the Audit Settings dialog.
5. Close the Audit View dialog.

Make Changes

Change and save your project; for example:

- Add a new package
- Add a new Class
- Add a new connector
- Change the name of an element
- Delete an element.

View Changes in the Audit View

Open the Audit View again (**View | Other Project Tools | Audit View**), and click on the **Refresh** (or **Load**) button to display a record of the changes you made.

13.2 Auditing Settings

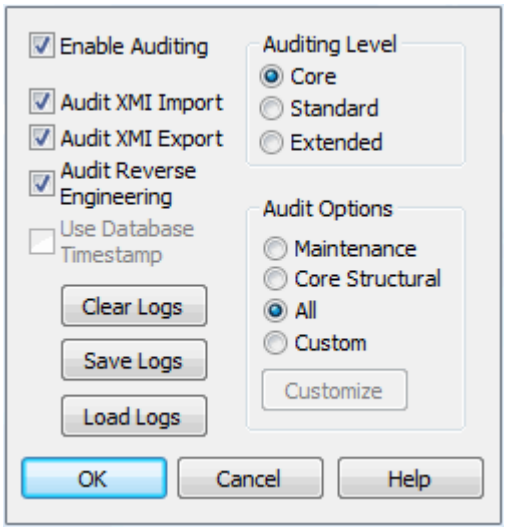
The Audit Settings dialog enables you to change what is recorded by the Auditing facility.

Note:

If security is enabled, you must have **Audit View** permission to turn Auditing on, and **Audit Settings** 183 permission to change Audit settings.

To open the Audit Settings dialog:

1. Select the **View | Other Project Tools | Audit View** menu option to open the Audit View.
2. Click on the **Audit Settings** button. The Audit Settings dialog displays.



The settings on this dialog are described in the following topics:

- Audit Scope 199
- Audit Logs 199
- Auditing Level 200
- Audit Options 200

13.2.1 Audit Scope

The Audit Settings dialog provides checkbox options for turning Auditing on, and for including or excluding areas of processing in Enterprise Architecture.

- **Enable Auditing** - select this checkbox to turn the Auditing facility on
- **Audit XMI Import -** select this checkbox to include XMI importing in the audit
- **Audit XMI Export -** select this checkbox to include XMI exporting in the audit

Note:

As version control uses XMI, these options must be selected to record changes from checking out packages.

- **Audit Reverse Engineering -** select this checkbox to include reverse engineering in the audit
- **Use Database Timestamp** - select this checkbox to use the database server's timestamp instead of each user's local timestamp; this improves security.

Note:

The **Use Database Timestamp** option is not available for projects stored in .EAP Files.

13.2.2 Audit Logs

The Audit Settings dialog enables you to administer your audit records. As the number of records increases, the performance of the Audit View 201 reduces. It is recommended that audit records that are not regularly required are saved to file, then cleared from the project. This helps ensure high performance.

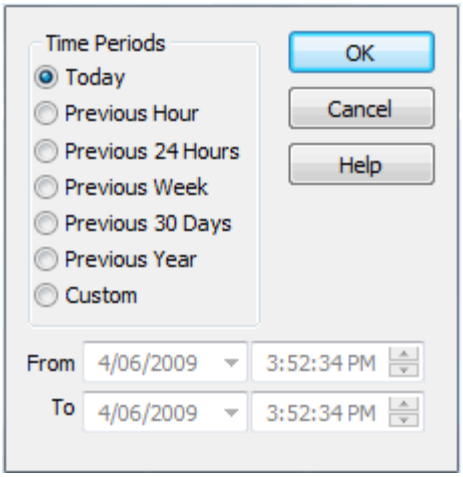
- **Clear Logs** - removes all log data from the current project; all data is permanently deleted. To keep the audit records outside the project, click on the **Save Logs** button to save the records before clearing them from the project
- **Save Logs** - saves a copy of the log items currently held in the database; these items remain in the database. To remove the items after saving the copy, click on the **Clear Logs** button
- **Load Logs** - enables you to load a previously saved set of logs back into the project. The file is not deleted by this operation. If duplicate logs exist in both the project and the file, these are skipped.

Note:

Some of these functions can be accessed through the Automation Interface. For more information, see the *Repository* topic in *SDK for Enterprise Architect.*

If you save or clear the log items, Enterprise Architect prompts you to specify whether to save or clear the items covering a specific period of time.

- If you click on the **No** button, you save or clear all log items currently held in the database.
- If you click on the **Yes** button, the Time Filter dialog displays, on which you select a standard time period or define your own.



13.2.3 Auditing Level

The Auditing Level panel provides three options that determine what kind of model changes are recorded.

- **Core** - select this radio button to record changes to elements (including attributes and operations), packages, connectors and some model-level information
- **Standard** - select this radio button to record the same changes as the **Core** option, plus changes to diagrams
- **Extended** - select this radio button to record the same changes as the **Standard** option, plus changes to security.

13.2.4 Audit Options

The following elements are always audited:

- Packages
- Notes
- Boundary
- Text
- Diagrams (if on *Standard* 200 level)
- Security (if on *Extended* 200 level)

The audit options enable you to configure auditing to record changes to only certain types of elements.

- **Maintenance** - select this radio button to audit maintenance elements; that is:
  - Package
  - Requirement
  - Feature
  - Use Case
  - Actor
  - Note

  - Issue
  - Change
- **Core Structural** - select this radio button to audit maintenance elements plus some structural elements; that is:
  - Package
  - Class
  - Interface
  - Signal
  - Node
  - Component
  - Artifact
  - Part
  - Port
  - Device
- **All** - select this radio button to audit all elements
- **Custom** - select this radio button to audit element types that you specify.

If you select the **Custom** option, the **Customize** button is made available. Click on this button to display a list of element types, and select the checkbox against each element type to include in the audit (or click on the **Select All** button to select every element type). Click on the **OK** button to save the selection.

Note:

Connectors are audited when they are connected to an element that is included in the Audit Options.

13.3 The Audit View

The Audit View provides an interface to the information that has been recorded by auditing. Open the Audit View by clicking on the:

- **View | Other Project Tools | Audit View** menu option.

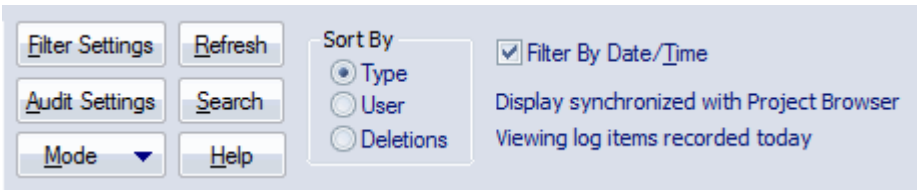
The Audit View is divided into three main areas:

- View controls
- Audit tree
- Record display.

The data in the Audit tree and Record display is determined by the view controls and mode 204 and, if synchronizing 205 with the Project Browser, by the package, diagram or element you have selected.

View Controls

The view controls provide a variety of settings for controlling auditing and the display of audit records.



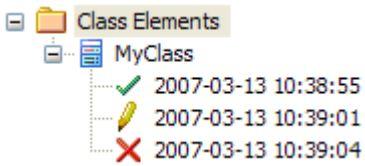
For information on these controls, see Audit View Controls 203.

Audit Tree

The audit tree displays the logs that have been recorded by auditing. What is displayed in the tree is affected by the settings of the View Controls 203 section, such as:

- Sorting
- Filter (by time)

- Mode
- Auditing settings 198 (what was actually recorded).



In the audit tree:

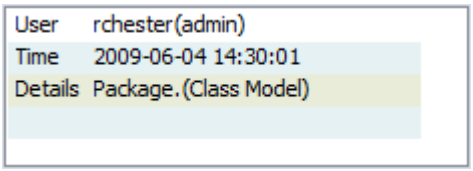
- The green tick indicates a creation
- The yellow pencil indicates an edit
- The red cross indicates a deletion.

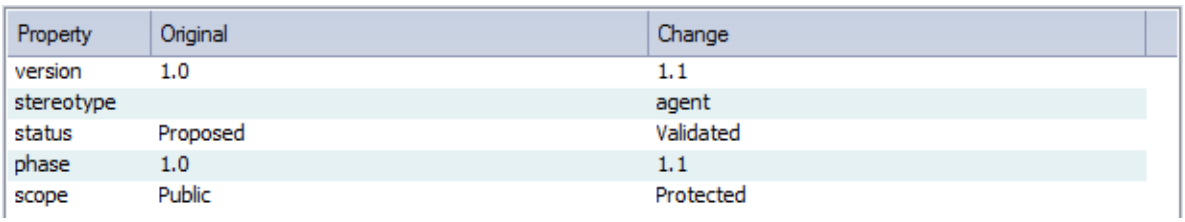
If you right-click on an element in the audit tree (such as *MyClass*) a context menu displays. This menu enables you to locate the selected element in:

- The Project Browser
- Any diagrams in which it exists.

Record Display

The record display is in two parts: the identity of the selected change, and the actual change made.





Property	Original	Change
version	1.0	1.1
stereotype		agent
status	Proposed	Validated
phase	1.0	1.1
scope	Public	Protected

Identity

The identity of a change consists of:

- The Windows username of the user that made the change

  Note:

  If security is enabled, the name is of the format: *WindowsUsername(SecurityUser).*

- When the change was made
- The *path* of the change (for example, Class *Class1* - Attribute *Att1* - Attribute Constraint *Constraint*).

Changes

The changes are displayed in a table format, showing the Property (or data item) name, its original value before the change and its value after the change.

If you double-click on an item in the list of changes (or right-click and select the **Show Differences** context menu option) the Difference dialog displays. This shows the specific changes that have been made, highlighting the edited, created or deleted characters.

13.3.1 Audit View Controls

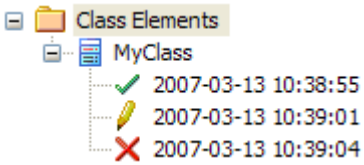
The Audit View controls provide a variety of settings for controlling auditing and the display of audit records.

- The **Load** (Advanced or Raw 204 modes) or **Refresh** (Standard 204 mode) button reloads the Audit Tree, updated with any new audit results.
- The **Search** button enables you to search through log items for a particular area. Log Items can be searched by Name, Type or GUID. The items are loaded and filtered with the current **Sort By**, **Time Filter** and **Mode** settings. If you refresh the Audit View, you must run the search again.
- The **Audit Settings** button opens the Audit Settings 198 dialog.

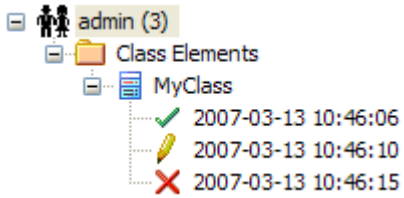
Sort-by Panel

The Sort By panel enables you to select one of three display settings:

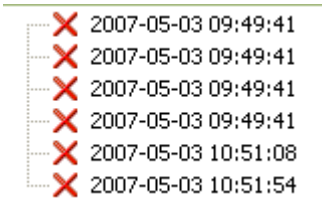
- **Type** - changes are grouped under element type (such as Class or Requirement), and then grouped under the changed element.



- **User** - changes are grouped under user name. The number of changes for each user is also displayed. Under each user name, changes are grouped as for the **Type** sort.

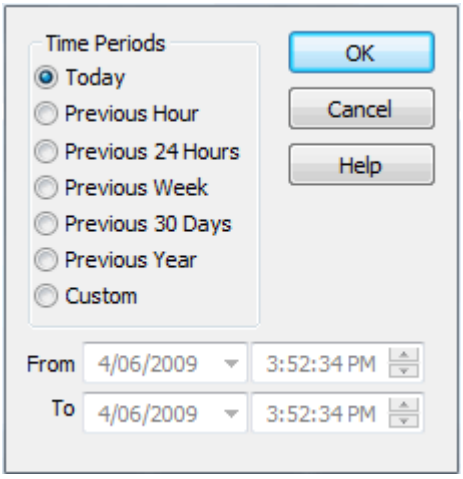


- **Deletions** - displays only deletions, shown in chronological order. If used with the **Search** button, this can be useful for recovering information on missing elements.



Filter by

The **Filter By Date/Time** checkbox enables filtering by time periods, which you set using the Time Filter dialog; click on the **Filter Settings** button to display this dialog.



Select:

- **Today** - to display changes occurring today
- **Previous Hour** - to display changes occurring in the last 60 minutes
- **Previous 24 Hours** - to display changes occurring in the last 24 hours
- **Previous Week** - to display changes occurring in the last 7 days
- **Previous 30 Days** - to display changes occurring in the last 30 days
- **Previous Year** - to display changes occurring in the last 365 days
- **Custom** - to define your own time period, using the **From** and **To** fields.

Note:

The six pre-configured time periods automatically update when you click on the **Refresh** button. Custom periods are static and do not automatically update.

Changes that occur *outside* the filter period you select are not shown in the Audit View. Once you have set a filter period, it remains set if you deselect the **Filter By Date/Time** checkbox. The Custom time period, too, is retained so that you can re-use it or modify it later if required.

Status Text

Beneath the **Filter By Date/Time** checkbox, status text displays to show which mode 204 has been selected, and which time filter 204 is being applied to the data.

Mode Button

The **Mode** button enables you to change the mode of the Audit View. The button displays a drop-down menu from which you can select:

- **Standard** - to interact with the Project Browser
- **Advanced** - to load large sets of log items

  Note:

  When in **Advanced** mode, a special Audit Settings group can be displayed in the Audit Tree 201. This records when Auditing has been enabled and disabled, along with who made the change, and the date and time of the change.

- **Raw** - to display all audit records without sorting (although any search and filtering you define still apply). Additional database information is displayed; this additional information might be unimportant.

Standard Mode

In **Standard** mode, Auditing is automatically synchronized with the Project Browser.

When synchronized, and where changes have been made, the Audit View reflects your selection from the Project Browser. If you click on:

- An element, the Audit View displays the history for that element
- A package, the Audit View displays the history for that package and its immediate children (but not the contents of nested packages)
- A diagram, the Audit View displays the history for that diagram and its contents (which could be drawn from a wide area of the Project Browser).

Advanced Mode

In **Advanced** mode, you can load sets of audit data independent of the Project Browser. These sets of data display all significant changes, but you can reduce the selection by filtering by time or by running a search. **Advanced** mode also displays:

- Changes to the Audit Settings
- When Audit Operations are executed
- Security changes (which can be browsed in the same way as other changes).

Raw Mode

In **Raw** mode, all data recorded by auditing is displayed in chronological order. This enables you to see a progression of changes, which can be especially useful in determining date-time inconsistencies. Search and filters can still be applied, enabling you to view all of today's changes in order, or all changes for a particular element in order, or both.

Note:

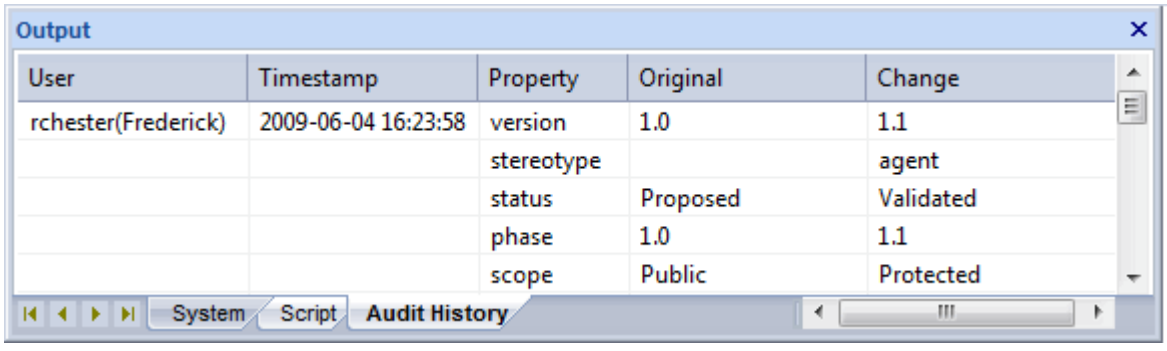
Some information displayed in **Raw** mode might be insignificant or only in machine-readable format.

13.4 Audit History Tab

When Auditing is turned on, an Auditing History tab is enabled in the Output window. To see this tab, you must have one or both of the Auditing 197 and Element List views open.

Note:

If security is enabled, you must have **Audit View** 183 permission to display data on the Audit History tab.



The information in the Audit History tab provides a history of changes to whichever element or connector you have selected in the:

- current diagram
- Project Browser
- Audit View, or
- Element List.

As you select different elements or connectors, the Audit History automatically updates to reflect your current selection. The information shows, for each change made to the element or connector:

- Who made the change
- When the change was made
- Where the change was made
- The value of the characteristic before the change
- The value of the characteristic after the change.

13.5 Auditing Performance Issues

Impact of Auditing on Other Facilities

Enabling auditing on a project increases the time taken for most actions. For most modeling tasks, this increase is insignificant; however, there are some situations where the difference is more substantial.

Large Deletions

Deleting large packages or package structures, or large numbers of elements, takes significantly longer with auditing on. You might disable auditing 198 before performing such a deletion.

XMI Imports

Importing XMI 106 takes longer with auditing enabled. A project-level option 198 is provided for disabling auditing of XMI Imports.

Reverse Engineering

Reverse engineering code takes longer with auditing enabled. A project-level option 198 is provided for disabling auditing of reverse engineering.

13.6 Audit View Performance Issues

Most operations in the Audit View are affected by the volume of use of the database - both by other facilities and by auditing itself. Some potential problems and their solutions are outlined below.

Navigating the Project Browser Within Auditing is Slow

- Try setting the time filter 204 to a period in the immediate past, such as **Today**, **Previous 24 Hours** or **Previous Week**. This time period updates each time you open or refresh the Audit View.
- Save log items 199 outside the project with the **Save Logs** button. If you then clear the logs you have just saved, the load time of the Audit View is reduced. You can reload logs into the project at any time, using the **Load Logs** button.

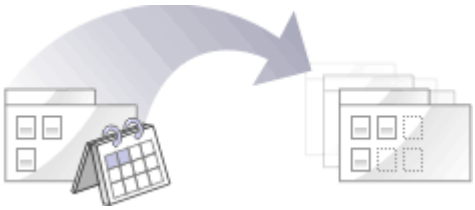
Navigating the Audit Tree is Slow

- Close the Audit History 205 tab in the Output window.

The Audit View is Slow in Loading and Changing Modes

- Try setting the time filter 204 to a period in the immediate past, such as **Today**, **Previous 24 Hours** or **Previous Week**. This time period updates each time you open or refresh the Audit View.
- Save log items 199 outside the project with the **Save Logs** button. If you then clear the logs you have just saved, the load time of the Audit View is reduced. You can reload logs into the project at any time, using the **Load Logs** button.

14 Baselines, Differencing and Merges



Enterprise Architect (Corporate, Business and Software Engineering, System Engineering and Ultimate editions) includes tools to help you manage and review changes to your models over time. These tools apply the concepts of *Baselines*, *Differencing* and *Merges*.

You can also obtain a snapshot of selected items in the model, using the Model View facility. In the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions of Enterprise Architect, this facility enables you to automatically generate this snapshot at intervals and, if there are changes in the items collected by the defined search, to trigger a notification to you of such changes. This enables you to monitor work flow 216 and other events of concern to you.

Baselines

Enterprise Architect (Corporate, Business and Software Engineering, System Engineering and Ultimate editions) provides a facility to create a Baseline 208 or snapshot of the contents of a selected package and its child packages at a particular point in time, enabling you to later compare that branch of the model at that time with the current state of the branch. Baselines are stored in the same XML format as is used for version control, but are stored *within* the project in compressed format. You can also have parallel copies of parts of your model for team development, and create Baselines within each copy to merge changes into the project master.

Differencing

Differencing (*Diff*, or *Compare* 210) enables you to explore the differences between the current state of a specific part of your project, and previous or parallel versions captured in a Baseline or an XMI 1.1 file on disk.

Merges

Once Differencing is complete, you can merge information from the Baseline into the current project; it is not possible to go the other way. You can merge information manually, change by change, or automatically by electing to merge in all changes in one batch procedure. You can also revert completely to the original Baseline by importing the stored XMI directly, and merge in information and elements from a Baseline in a different project, making it possible to keep multiple versions of a single model in synch.

The merge options 213 are available on the Compare Utility tab, which shows the results of a comparison. The options are available through the toolbar, context menus and the keyboard.

Notes:

- You use Baselines, Differencing and Merges essentially to compare two snapshots of a specific part of your project, to capture the differences between them and either roll back or incorporate selected changes or all changes. Enterprise Architect Corporate, Business and Software Engineering, System Engineering and Ultimate editions have another facility, Auditing 197, which you can switch on to perform continuous monitoring of changes across the project. You can dovetail your use of each facility to meet the range of your change management requirements.
- If a package under version control forms part of a Baseline, and that package is checked in to the model, you cannot merge the original data from the Baseline into that package.
- If security is enabled you must have **Manage Baselines** 183 permission to create, import and delete Baselines, and **Restore From Baseline** permission to merge data from a Baseline. Security permissions are not required to select an existing Baseline and perform a comparison with the model.

14.1 Baselines

Enterprise Architect (Corporate, Business and Software Engineering, System Engineering and Ultimate editions) provides a facility to 'Baseline' (snapshot) a model branch at a particular point in time for later comparison with the current package state. This is most useful for determining changes made to the model during development compared to some Baseline saved at a crucial point - for example the completion of a phase or version iteration. Baselines are stored within the model in compressed XML format. More than one Baseline can be stored against a single Enterprise Architect package.

You can also save a Baseline to an XML file for storage or archive, or for distributing to other users working on models derived from a master project.

Baselines are particularly useful during requirements management to check for changes, additions and deletions that have occurred since the start of the current work phase. Knowing how a model has changed is an important part of managing change and the overall development process.

Baselines are generally used in conjunction with the *Compare utility (diff* 210), which is built into the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect.

A typical scenario for using Baselines 208 would be to:

1. Create 210 the base model branch to a sufficient point to create a Baseline (checkpoint). Create and store the Baseline as Version 0.1a.
2. As work continues on development, managers and developers can check the current model branch against the Baseline for important modifications, additions and deletions. The Compare (diff) tool can be invoked from the Baseline dialog to check the current model branch against the stored version.
3. As required, minor Baselines can be created to check recent progress. These 'temporary Baselines' are useful for managing change when a lot of work is being done and it is important to only see what has changed in, for example, the last 24 hours.
4. At sign-off or the move to a new version/phase, a major Baseline can be created to capture the new state of the model. Minor Baselines created earlier can be deleted if required to save space.

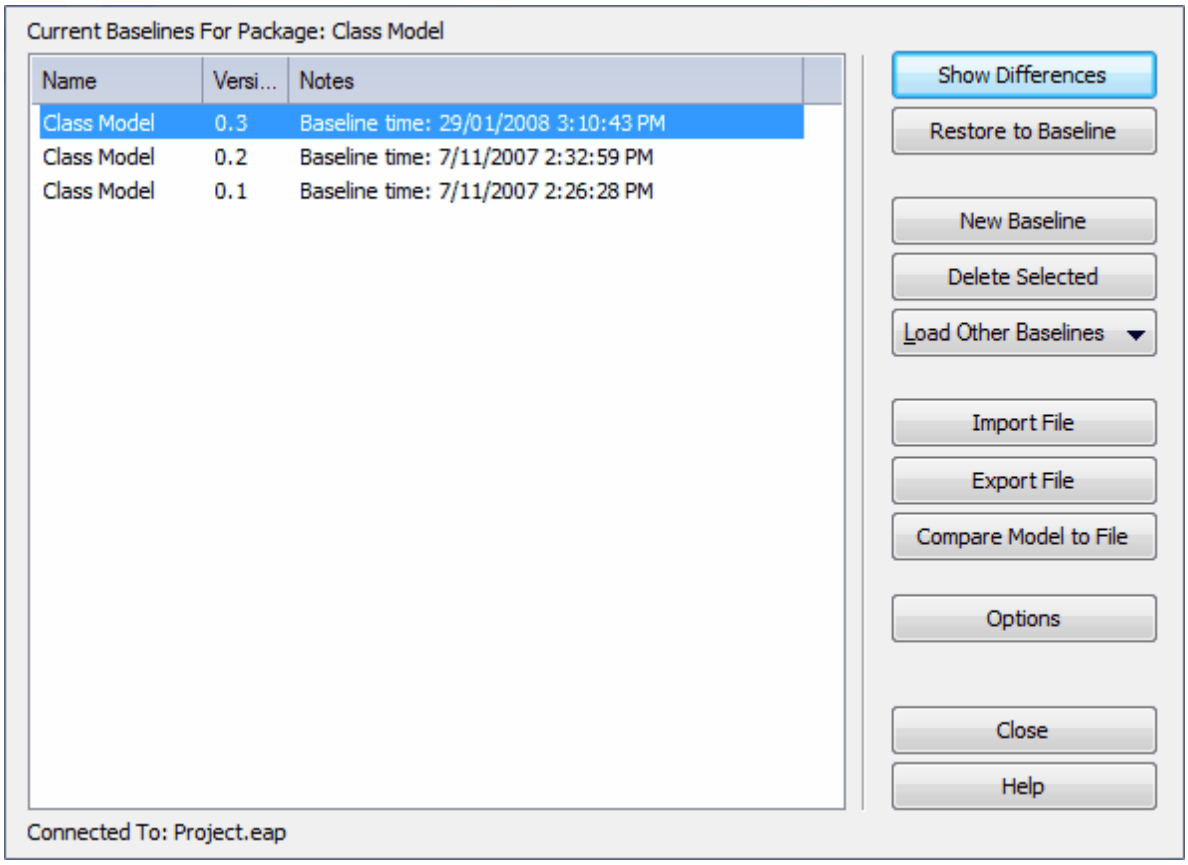
Important Considerations

- Baselines are based on the GUID or unique ID of a particular package. Enterprise Architect checks for that ID as the root element within the XML document being used as a Baseline. When you export a package to XML, the package you export is the root element. Likewise when you create a Baseline, the current package is the root package of the XML Baseline. When you save information in a version control system, the current version controlled package is again the root package of the document.
- It is not useful to create a Baseline by importing an XMI package file created by version controlling a package that itself contains version-controlled child packages. That type of XMI package file contains stubs for the child packages, not full information on the child packages and elements.
- XML files must be in the same format used by the Baseline engine - currently the **UML 1.3 XMI 1.1 format** (plus Enterprise Architect extensions), which contains all the information necessary to reconstruct a UML model, even a UML 2.x model.

14.1.1 Manage Baselines

Enterprise Architect provides a range of facilities for working with and managing Baselines, through the Manage Baselines dialog.

To open this dialog, right-click on the package at the head of the appropriate model branch and select the **Package Control | Manage Baselines** context menu option.

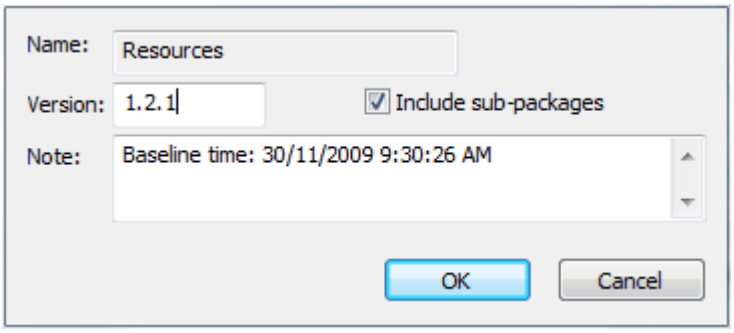


Option	Use to
Current Baselines For Package: <Name>	Review the baselines for the current model branch, listed by version reference 210 with the highest alphabetical/numerical value at the top. If an entry is longer than the display area, a horizontal scroll bar displays at the bottom of the panel. Use this to scroll to the text that is not shown.
Show Differences	Run the Compare 210 utility on the selected Baseline and the current model branch, and to display the differences 212 between the two.
Restore to Baseline	Completely restore the model branch from the selected Baseline.
New Baseline	Create a new Baseline 210.
Delete Selected	Delete the selected Baseline.
Load Other Baselines	Display a drop-down menu that enables you to load Baselines from another model, in either an .EAP file or a DBMS file. • For .EAP files, a browser displays; locate the required project file • For DBMS files, the Windows Data Link Properties dialog displays; select the data provider and click on the **OK** button to display the Select Data Source dialog, from which you select the required project. In either case, the *Connected To:* message at the bottom of the Manage Baselines dialog changes to the name of the alternative model. To return the dialog to the original project, select the third option on the drop down list: **Load From Selected Package**.
Import File	Import an XML 1.1 file from the file system as a new Baseline for this current model branch.

Option	Use to
Export File	Export the selected Baseline to an XML file on disk.
Compare Model to File	Compare the selected model branch with an XML **1.1** file on disk. A browser displays, which you use to locate the file.
Options	Set filters 211 to make the comparison more specific.

14.1.2 Create Baselines

Open the New Baseline dialog by clicking on the **New Baseline** button on the Manage Baselines 208 dialog.



Option	Use to
Name	Display the package name of the currently selected model branch.
Version	Type a unique version reference for this Baseline. This can consist of any alphanumeric characters. The Manage Baselines dialog sorts the Baselines according to the value of this field.
Include sub-packages	Include the entire sub-package hierarchy of this branch in the Baseline. Defaults to selected. If you deselect the checkbox, only the immediate contents (XMI stubs) of the package are included in the Baseline.
Note	Edit the default current time and date to any other value. The field is a single-line entry, for display on the Manage Baselines dialog (a one-line-per-entry list).

Click on the **OK** button to create a new Baseline and return to the Manage Baselines dialog.

14.2 The Compare Utility (Diff)

Notes:

- This utility is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect.
- You cannot compare the current model with an XMI 2.1 file; the utility can only compare with an XMI 1.1 file.

Enterprise Architect has a comprehensive and powerful differencing utility built in. This utility enables you to compare a model branch in Enterprise Architect with:

- A Baseline created using the Baseline functionality (Corporate, Business and Software Engineering, System Engineering and Ultimate editions)
- A Baseline stored in a *different* model
- An XML 1.1 file on disk created previously using the Enterprise Architect XML export facility (user selects

file), or

- The current version-controlled XMI 1.1 file on disk as created when using Version Control in Enterprise Architect (file automatically selected).

Compare (*diff*) lets you explore what has changed within a model over time and how previous versions of a model branch differ from what is currently in the model. It is even possible to do a full model comparison by exporting all of Model A to XMI, then using **Compare Model to File** from within the current model (Model B).

Comparing and checking model development at various points in the process is an important aspect of managing change and development, keeping track of what is being modified and ensuring the development and design process is on track.

Access to the *Compare* utility is available from:

1. The Baseline dialog; from the Project Browser context menu, select the **Package Control | Manage Baselines** option.
2. The Project Browser context menu; select **Package Control | Compare with XMI File** (for a package not under version control).
3. The Project Browser context menu; select **Package Control | Compare with Controlled Version** (for a package under version control).

Differencing With Baselines

As a Baseline is stored within a model and contains all the information, elements and connections for a package at a point in time, it can be used within Enterprise Architect to track changes to model elements over time. The Differencing engine first builds a representation of the current package in memory, based on what is currently in the model. It then compares this with the stored Baseline, highlighting changes, new elements, missing elements and elements that have been moved to other packages. It is possible to filter the resultant output to display only one particular kind of change: for example, additions to the model.

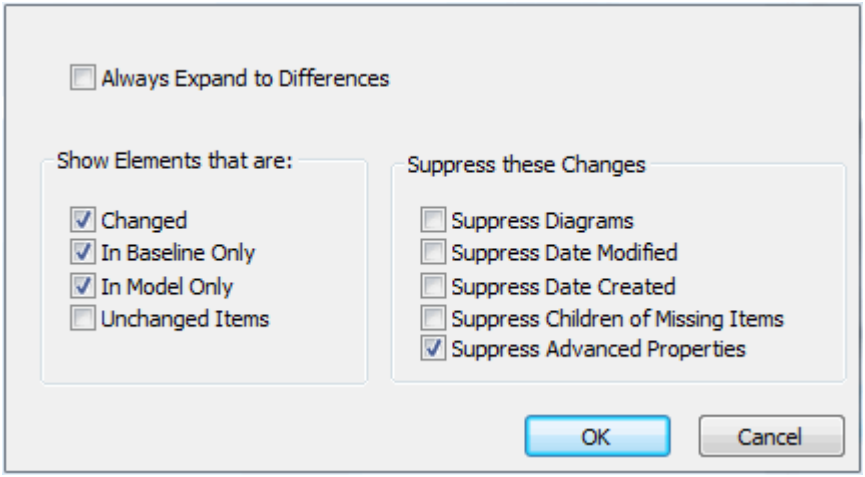
If a Baseline has been created to ignore child package content, a comparison between that Baseline and the model does not include any child package content in the model.

See Example Comparison for an example of a model comparison.

14.3 Compare Options

The Compare Options dialog enables you to refine the output of the Compare utility when it compares the current model with a Baseline. To display the dialog, either:

- Click on the **Options** button on the Manage Baselines dialog, or
- Click on the Compare Options icon on the Compare Utility tab toolbar.



Option	Use to
Always Expand to	Always display the list of elements fully expanded to show changes.

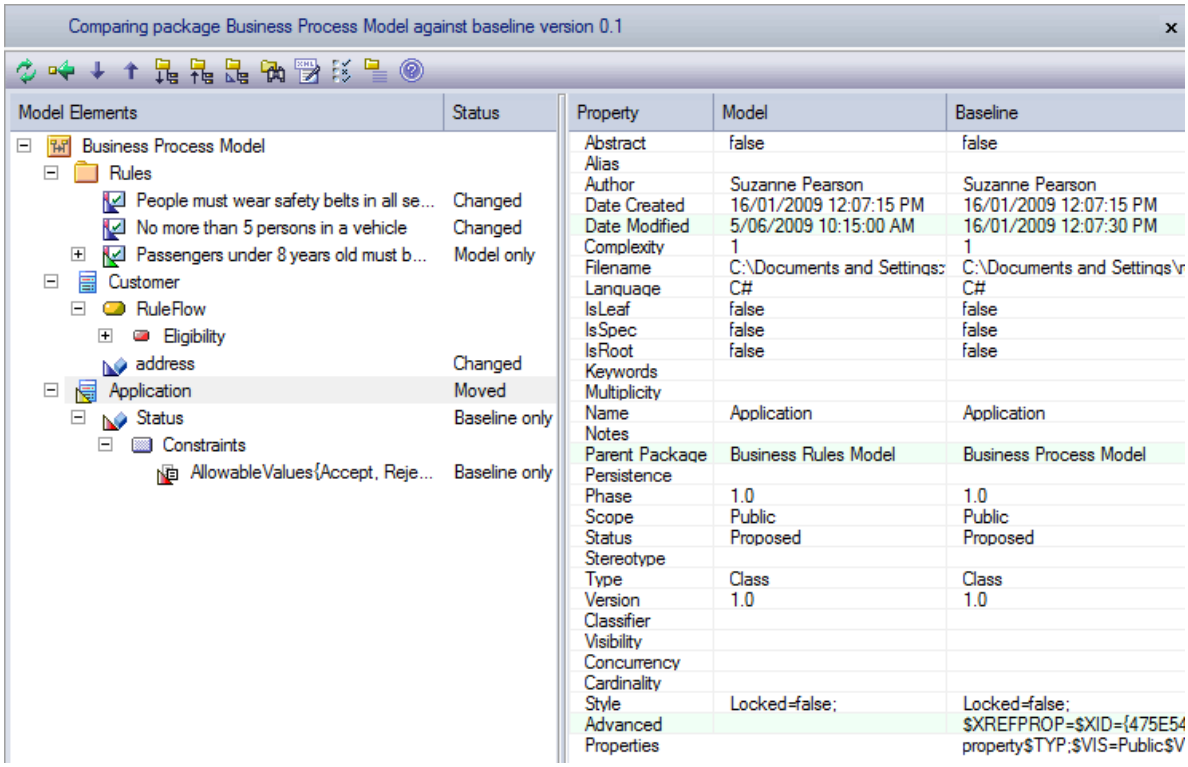
Option	Use to
Differences	If you deselect the checkbox, when the Compare Utility tab is first opened it lists the package contents to element level, and you expand each element as required to show the changed items. For large branches of the model, it is better to leave the checkbox unselected.
Show Elements that are	List elements that: • Have been changed since the Baseline was created • Are in the Baseline only (that is, have been deleted from the model since the Baseline was created) • Are in the model only (that is, have been created since the Baseline was created) • Have not changed since the Baseline was created (you might generally leave this checkbox unselected).
Suppress these Changes	Exclude: • Changes to diagrams • Changes to the **Date Modified** field for an item • Changes to the **Date Created** field for an item • Child items of a deleted item • Changes to advanced properties (defaults to selected).

If the Compare Utility tab shows the results of a Baseline comparison, when you click on the **OK** button the display refreshes to refine the information according to the options you have selected.

14.4 Example Comparison

The diagram below shows the result of a comparison between a package (*Business Process Model*) in the current project and that package in a Baseline (*version 0.1*) captured at an earlier date. Notice that:

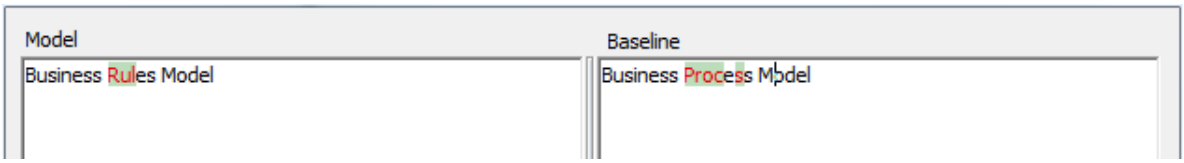
- The results are displayed on the Compare Utility tab.
- A hierarchy of model elements is displayed in the left-hand pane. It is clearly visible, from the **Status** column and from the triangular icons, which items in the hierarchy have been changed (Status *Changed*), deleted from the model (*Baseline only*), added to the model (*Model only*) and switched to different packages (*Moved*) since the Baseline was captured.
- If you click on an item with a **Status** entry in the left hand pane, the right-hand pane displays a table of properties showing the values of those properties in the current model and in the Baseline. For each property where there is a difference between the model and the Baseline, the row is highlighted. This example shows that the Class element named *Application* was moved from the *Business Process Model* package (as shown in the Baseline) to the *Business Rules Model* package on 5 June 2009 (as shown in the Model).



The Compare Utility tab enables you to perform operations (such as merging or rolling back changes) on the reported information, using the toolbar, context menu and keyboard 213.

Higher Level of Detail

The right panel of the Compare Utility tab might, for some fields, display only part of the value (such as **Advanced Properties**, above). It might also not be immediately obvious what a change is. In either case, you can double-click on the property to display full details and to highlight the exact differences. The following example shows the highlighted changes to **Parent Package**.



14.4.1 Compare Utility Tab Options

The Compare Utility tab enables you to perform operations on the reported information, using the toolbar, context menu and certain keyboard keys.

Note:

If a package under version control forms part of a Baseline, and that package is checked in to the model, you cannot merge the original data from the Baseline into that package.

Toolbar

The toolbar is at the top of the left-hand panel. The icons operate either on the comparison as a whole or on the currently-selected item in the left hand panel.

The toolbar icon names and functions are, from left to right:

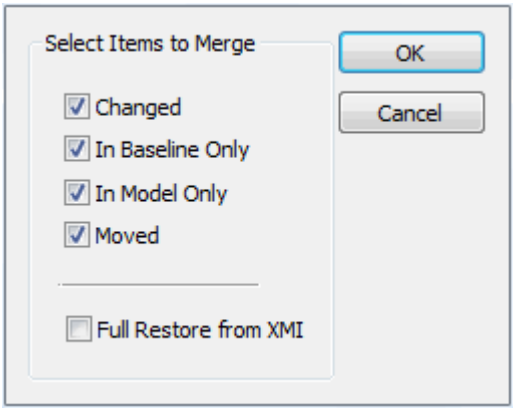
Option	Use to
Refresh	Re-run the comparison to refresh the current display.
Merge To Model	Merge the values of the currently-selected item in the Baseline back into the model.
Next Change	Highlight the next changed item (skips *Moved* items).
Previous Change	Highlight the previously-changed item.
Expand All	Fully expand the selected item.
Collapse All	Collapse the changed items in the selected item.
Expand To Changed Items	Expand the selected item to show changed items only (in the event that you have selected to also show 211 unchanged items in the comparison).
Find in Project Browser	Highlight the item in the Project Browser.
Log To XML	Log the changes to an XML file. A browser displays, on which you specify the file name and location.
Compare Options	Display the Compare Options 211 dialog.
Manage Baselines	Display the Manage Baselines 208 dialog.
Help	Display the Help topic *Baselines, Differencing and Merges* 207.

Context Menu

Each item in the hierarchy has a context menu, which you display by right-clicking on the item. The options displayed depend on the level of the item in the hierarchy, but include some or all of the following:

Option	Use to
Merge from Baseline **Add from Baseline**	Restore the item in the model to the Baseline state, or restore a deleted item from the Baseline.
Delete from Model	Remove a recently-created item from the model.
Merge From Baseline (with Options)	(For the root node of the hierarchy on the Compare Utility tab.) Display the Merge dialog (see below) which enables you to specify options for rolling back the whole model branch to the Baseline state.
Refresh	(Object-level items). Re-run the comparison to refresh the current display.
Find in Project Browser	Locate and highlight the item in the Project Browser.
Expand All	Fully expand the selected item.
Expand To Changed Items	Expand the selected item to show changed items only.

Option	Use to
Collapse All	Collapse the changed items in the selected item.
Log To XML	Log the changes to an XML file. A browser displays, on which you specify the file name and location.
Compare Options	Display the Compare Options 211 dialog.



Select the appropriate checkbox against each type of difference to roll back in the model from the Baseline.

Option	Use to
Changed	Restore all changed items in the model branch to the Baseline state.
In Baseline Only	Restore all deleted items to the model branch from the Baseline.
In Model Only	Remove all recently-created items from the model branch.
Moved	Restore all moved items to their original locations, as identified in the Baseline.
Full Restore from XMI	Completely restore the model branch to the version held in the Baseline XMI 1.1 file, (using the XMI Import function). (Automatically selects all the other options.)

Keyboard

You can also use the following keyboard keys to move up and down the hierarchy, or to roll back changes:

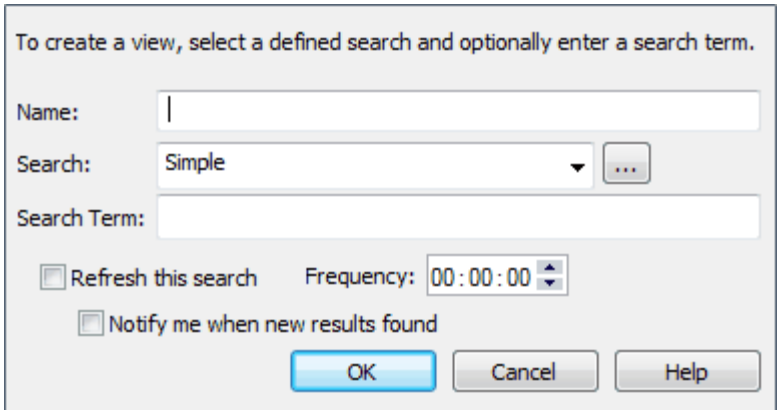
- **[Ctrl]+[↓]** - expand and highlight the next changed item
- **[Ctrl]+[↑]** - expand and highlight the previous changed item
- **[Ctrl]+[←]** - undo the changes for a selected item (roll back to the Baseline values).

15 Monitor Events

In the Enterprise Architect Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions, the Model Views facility enables you to:

- automatically refresh the search in a View at an interval that you define
- notify you if the results of the search change between two consecutive searches.

For information on the Model Views facility, see the *View Options* section of *Using Enterprise Architect - UML Modeling Tool*.

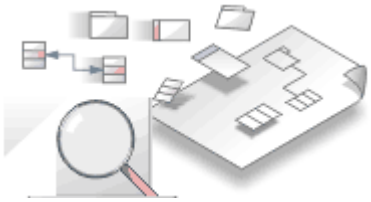


You can therefore use the Model Views facility to monitor various events in the development project, depending on how you set up the search in a View. You could, for example, set up a search to detect:

- Change items, or Issue items, so that Enterprise Architect would notify you as new items were created
- Element Status, Type, Phase, Version, Priority and/or date of last update, so that Enterprise Architect would notify you as items were progressed to fall in to the level of work represented by the search categories, and to move out of the categories into the next level of work.
- Tagged Values, so that - again - as items were changed to satisfy the criteria of a sequence of searches, the progression of items through a set of stages could be checked and managed.

People responsible for different stages in a process could have their own Model View searches so that as a development, validation or authorization task falls due the responsible person is automatically notified, and when the work is complete both the next person in line and the overseeing manager are notified.

16 Traceability



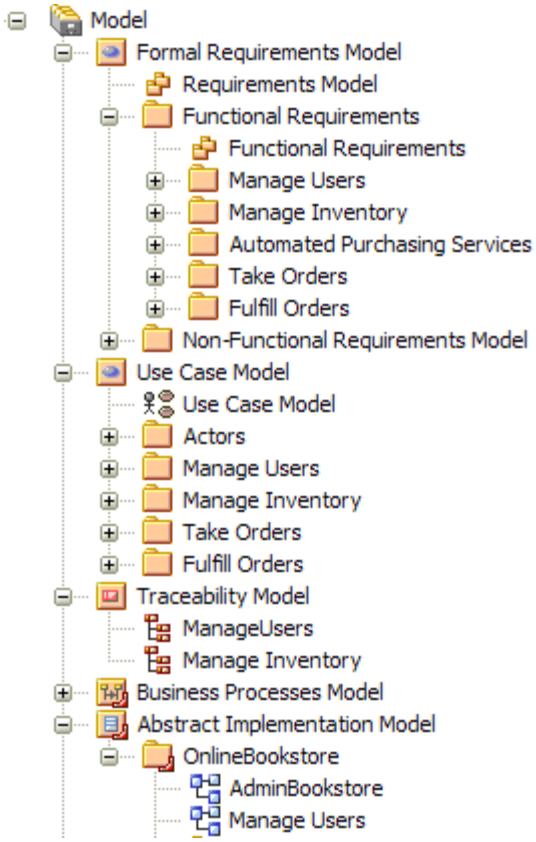
Traceability identifies the way a given process has been, or is to be, developed in a system. The process can be an internal, model-management process, where you monitor work by asking questions such as 'what work has been done to realize this Requirement or Use Case?', or a business or system process that is being modeled, where you ask questions such as 'what Requirements, Use Cases, Classes, Components, Test Cases and other elements define the implementation and deployment of this process?'

Traceability also helps to clarify the aspects of a process that the model does not address. A process typically includes a range of manual, automated and external procedures. A correctly-structured model illustrates exactly what requirements and functionality service a particular process; any missing functionality must come from other systems, developments or procedures.

There are various tools in Enterprise Architect that enable you to trace the definition and implementation of a process from initial requirement to generated code or technical deployment, or vice versa. If you have performed any Transformations in developing your model and code (see the *MDA Transformations User Guide*), Enterprise Architect automatically creates Transformation Dependency connectors that you can trace - with the Traceability window (see *Using Enterprise Architect - UML Modeling Tool)* - to establish what objects and code have been generated from each PSM element, or what the initial PSM element was for a generated object. Whether you use transformations or develop the stages of the model in other ways, you can build up a range of Traceability diagrams 222 (Custom diagrams) to identify the development pathway and the dependencies between entities such as Requirements, Use Cases, Classes, Packages, Test Cases and other model artefacts, or possibly between these entities and the overall business process model (see *Extending UML With Enterprise Architect*).

Maintaining Traceability

The following Project Browser hierarchy represents a model that is structured to enable traceability. Notice firstly that themes are developed in the structure - the Requirements Model, Use Case Model and Abstract Implementation Model each contain units with the same functional names, such as *Manage Users* and *Manage Inventory*. These units help you to quickly develop a *Traceability Model* within your project, to trace development and implementation.



Traceability in this model is examined further in the following topics:

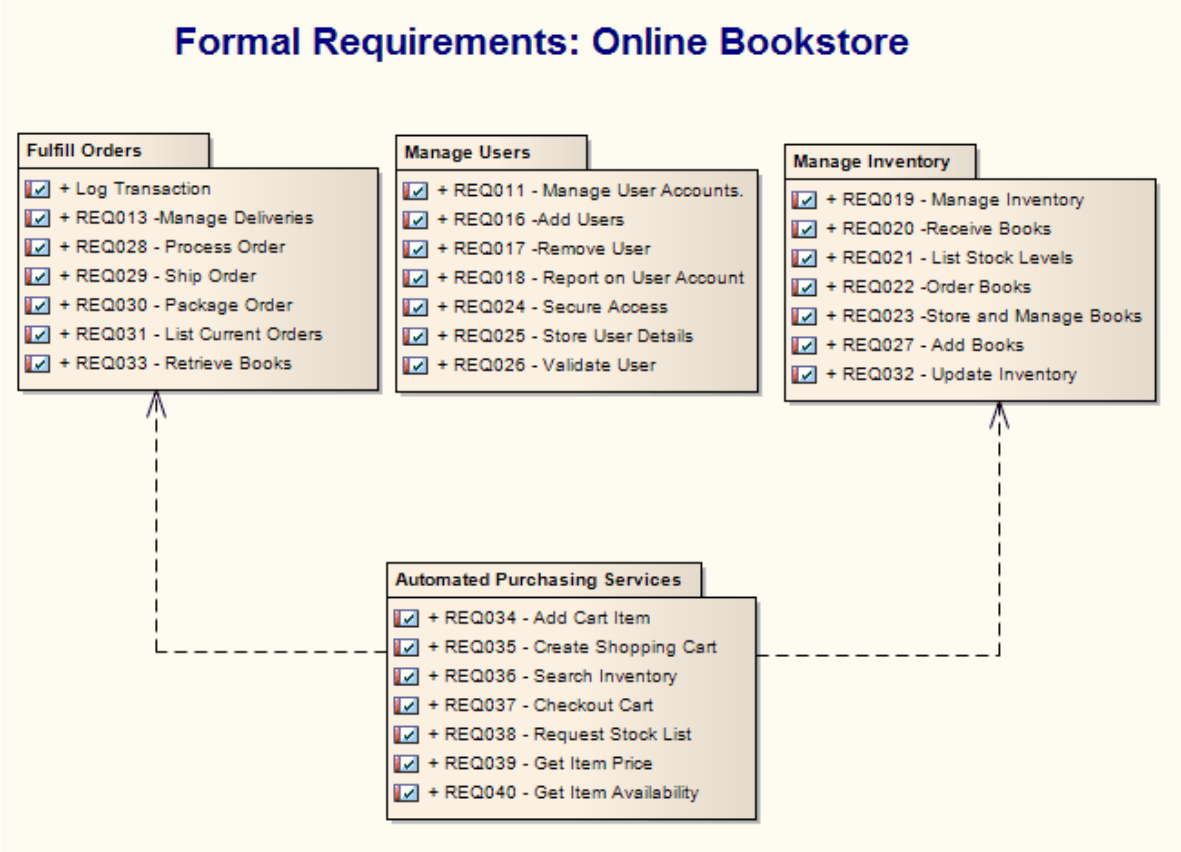
- Packages and Elements 218
- Create Traceability Diagrams 222
- Traceability Tools 224

16.1 Packages and Elements

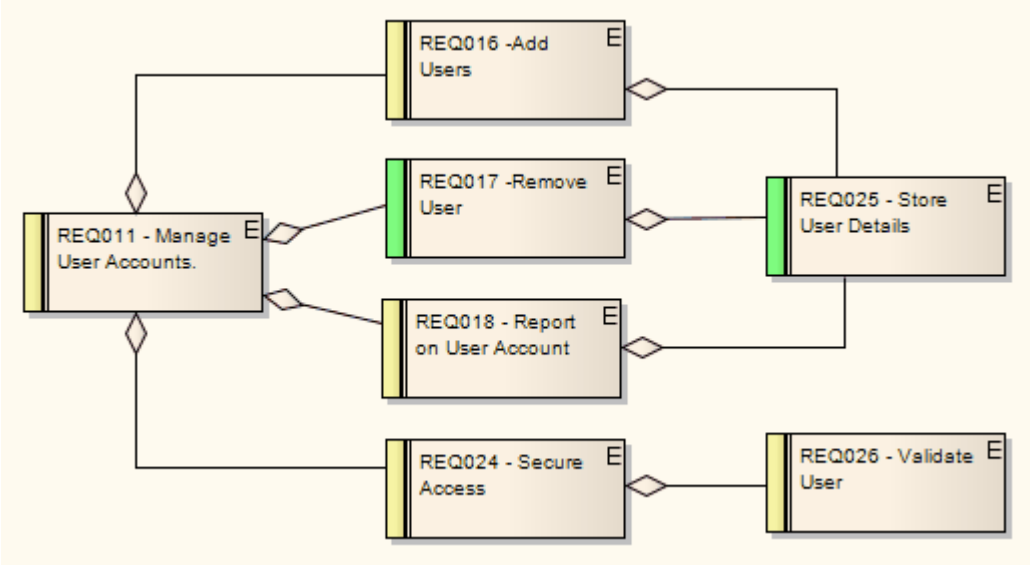
Analyzing the requirements of a system or process helps you identify units of functional activity in that system or process. You can represent these units with Packages, which you then populate with elements of relevance to the functional units and to the stage of development.

In the following four diagrams, you can see how Requirements and Use Case Packages are created to group the Requirements elements and Use Case elements that define and start to implement each of several functional areas. By maintaining the same structure in each model in the project (as discussed in the Traceability 217 topic), you also make it easy to trace the project development through the stages.

Requirement Packages for Online Bookstore Process

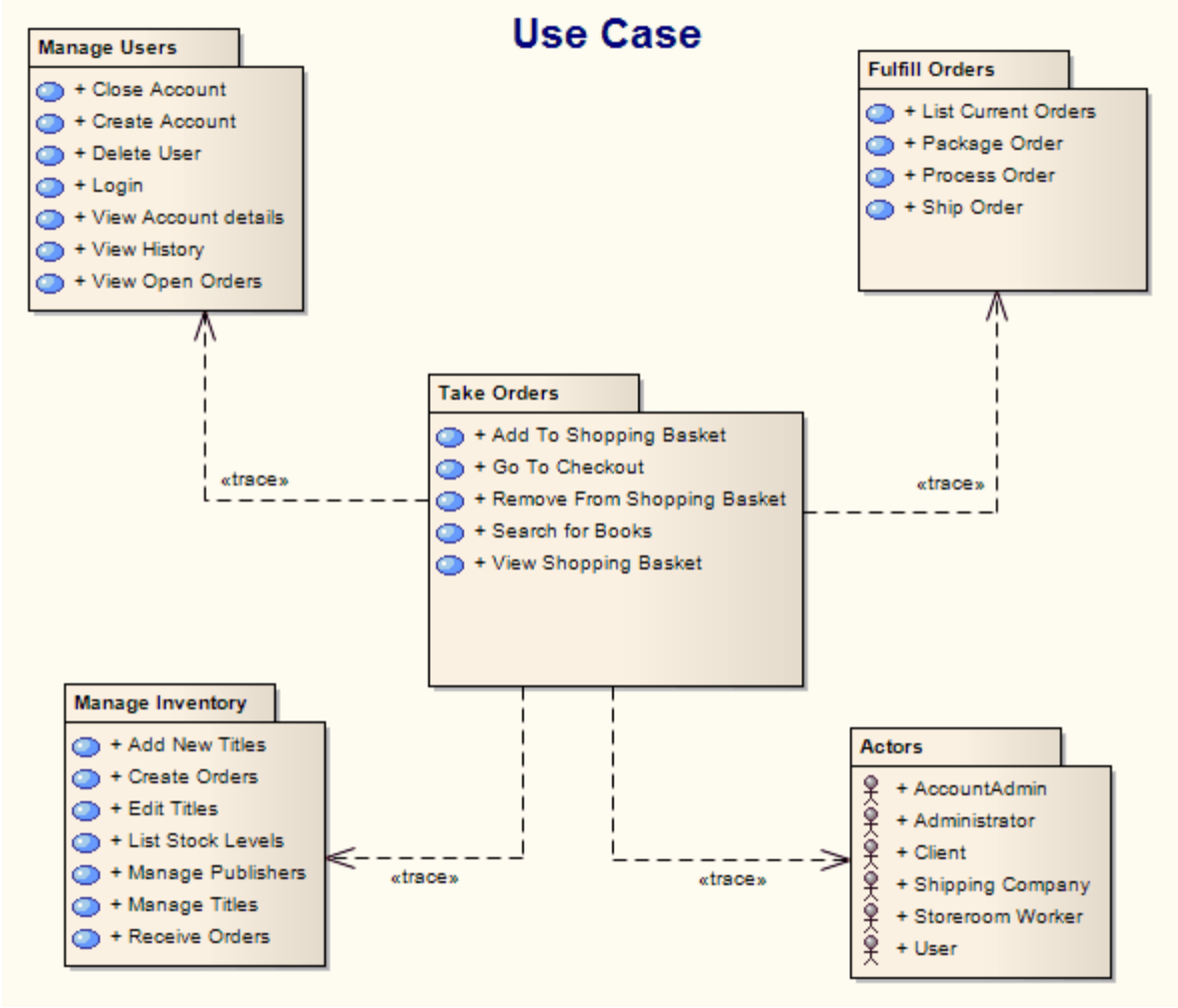


Requirements in Manage Users Unit of Online Bookstore Process

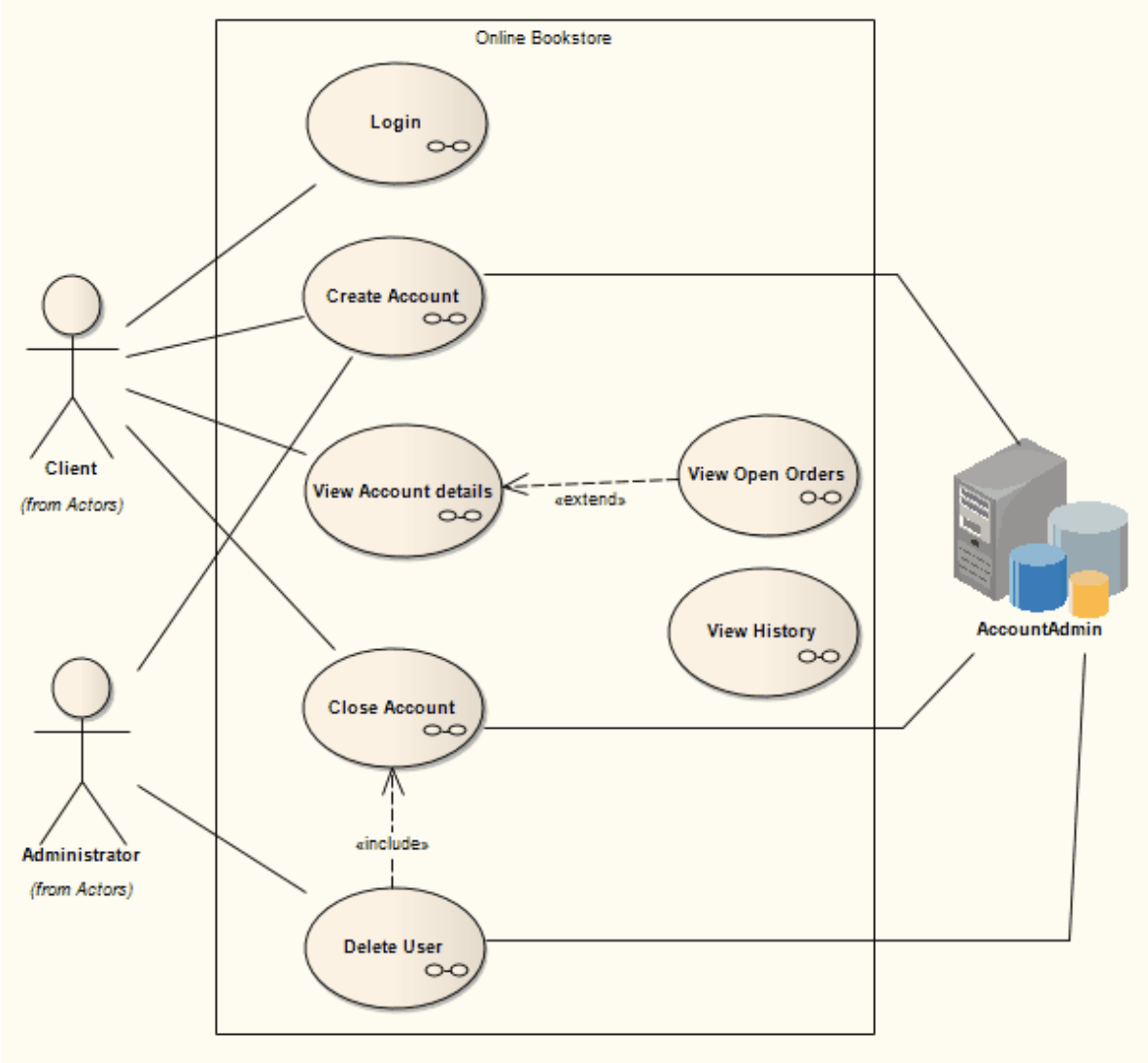


The Requirements diagram also makes it clear what Requirements form subsets of others, or are components of more than one other Requirement.

Use Case Packages For Online Bookstore Process



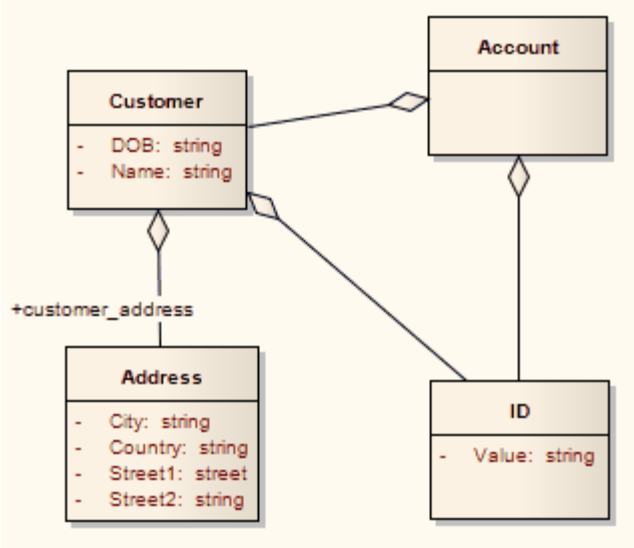
Use Cases in Manage Users Unit of Online Bookstore Process



The Use Case diagrams can also clarify what aspects of a process require or enable human intervention, and which require or enable system intervention.

Implementation Stage

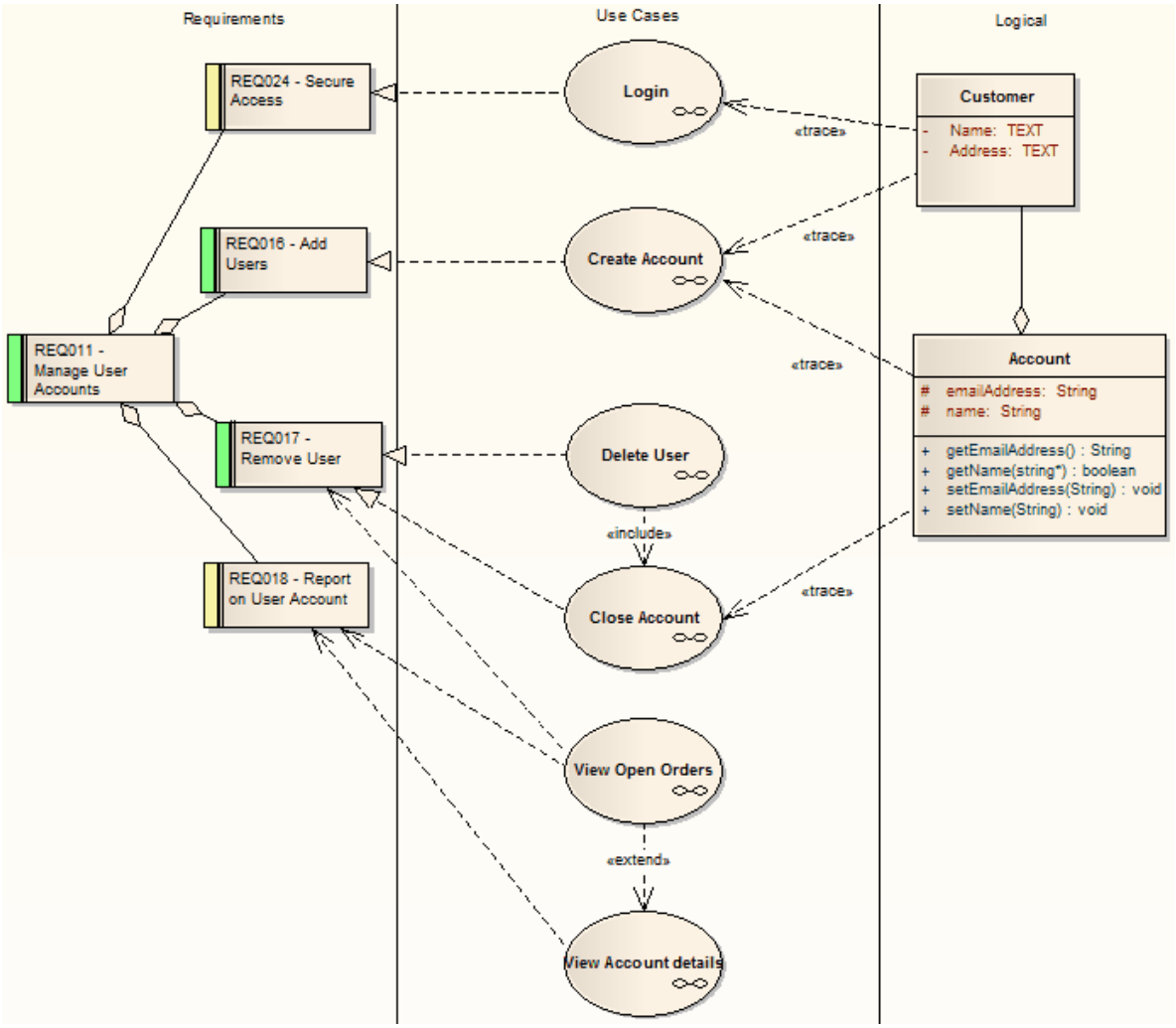
For completeness, you could also consider the next stage, the implementation of some of these Use Cases, as represented by Class elements associated with this functional unit.



16.2 Create Traceability Diagrams

Having structured the models of a project to indicate directions and theoretical relationships between the models and packages, you can formalize these directions on a *Traceability* diagram, using *Realize*, *Trace* and similar relationships (see the *UML Connectors* topic in the *UML Dictionary*).

You initially create a Traceability diagram as a Custom diagram (see the *UML Diagrams* topic in the *UML Dictionary*), but if you are creating the diagram manually you can use elements and relationships from other Toolbox pages to develop the diagram as broadly as is necessary.



You can also generate the diagram using the **Add | Related Elements** context menu option to automatically bring in elements linked to the selected element (see the *Work With Elements* section of *UML Modeling With Enterprise Architect - UML Modeling Tool*). It is probably better to add the elements in stages, one level at a time, but you could add several levels in one go to see how far the hierarchy extends and to identify relationship and element types to exclude from the 'clean' diagram. You could perform a similar operation, one element at a time, using the Relationships window (see the *Dockable Windows* section in *Using Enterprise Architect - UML Modeling Tool*).

The above diagram instantly shows how two levels of Requirements are realized by Use Cases, and which Requirement is realized by which Use Case(s). It also shows how some of the Use Cases are implemented by Class elements. Further, you can drill down on the Use Cases (or, in other Traceability diagrams, any other composite elements) to display more detailed diagrams showing how the Use Case meets the Requirement. The *Close Account* Use Case, for example, contains a Communication diagram and a Sequence diagram.

You can tailor your Traceability diagrams to depict any level of granularity and any stages of development that are appropriate. You might narrow the above diagram, for example, to show development from just the *Remove User* Requirement, and extend it to include Interfaces, Components, Test Case elements or any other facet of the system or process.

Whilst the Traceability diagram itself provides information on the definition, design and implementation of a business process feature, much more information can be obtained using tools 224 such as the Relationships Matrix and relationships Traceability window.

16.3 Traceability Tools

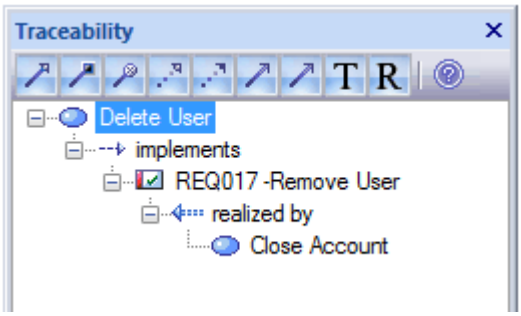
The model structure 218 and Traceability diagram 222 act as starting points for tracing the definition, design and implementation of a feature of a system or process. By applying tools such as the Relationship Matrix and Traceability window to these, you can follow threads throughout the model to determine how the feature is implemented and tested. You can also obtain information on what elements realize and are realized by the elements in a given package, using the Dependency report and Implementation report, respectively (see the *Other Documents* topic in *Report Creation in UML Models*).

Traceability Window

The Traceability window (see *Using Enterprise Architect - UML Modeling Tool*) is a most useful and versatile traceability tool. Starting with a Traceability diagram or a package structure in the Project Browser, you can use the Traceability window to quickly explore the relationship chain of which any element is a component. When you click on the element, it immediately becomes the top point in the Traceability window. When you click on the background of a diagram, all elements in the diagram are listed in the Traceability window, and you can follow the threads starting at each element through the diagram.

If you require a rapid, broad-brush view of relationship flows in the project structure, starting with a general list of - say - all functional Requirements, you can use a combination of Model Search, Project Browser and Traceability window; this is a powerful tool for scanning your project, identifying how elements have been organized, and how they interact. For example, the Model Search would list all the requirements, you could rapidly click on each element and immediately see in the Project Browser where it has been grouped, and at the same time - in the Traceability window - how that element interacts with other elements in the model.

You can select any or all of the relationship types available in the Traceability window toolbox. The single type selected below is *Realizes* (Implements), and the selected element is the *Delete User* Use Case. The Traceability window then shows that *Delete User* implements *REQ017*, but this is also partially realized by the *Close Account* Use Case.



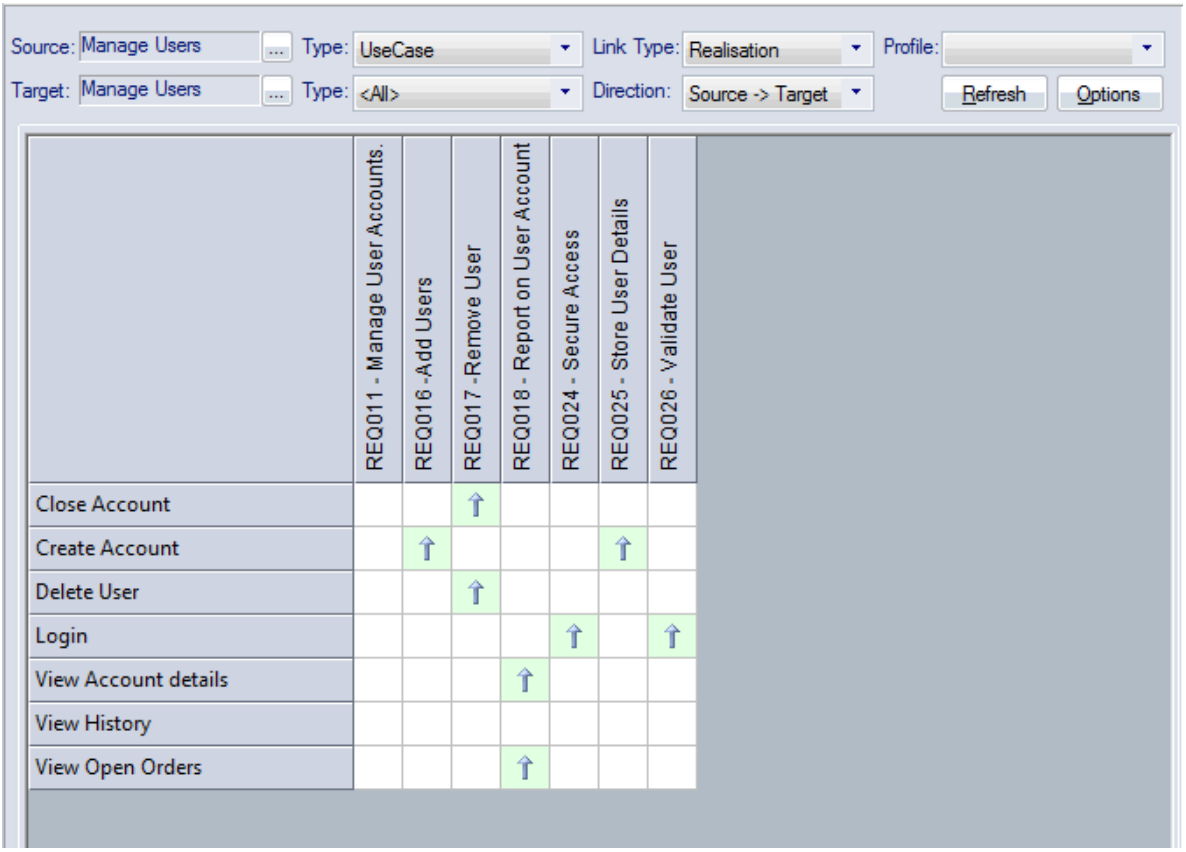
By moving the cursor around a diagram or the Project Browser, and/or changing the relationship type combinations in the Traceability window, you can quickly see how elements are connected and how they influence each other. For example, you can see that REQ017 is realized by two Use Cases, so you might then explore what else influences and is influenced by these two Use Cases. The Traceability window takes you well beyond what is likely to be depicted on any single diagram.

If you have used transformations to develop your model, the **T** icon displays the transformation dependencies that exist between an element in a PIM and elements in the PSMs.

Relationships Matrix

Using the Relationships Matrix (see *UML Modeling With Enterprise Architect - UML Modeling Tool*), you can both create and study the relationships between, for example, the Requirements and Use Cases for a module. You might identify the 'theme' package (in this case, *Manage Users*) in the Requirements model and the Use Case model as the source and target packages, and explore the likely element and connector types in the packages. This, like the Traceability diagram, identifies which Requirements are (or should be) realized by which Use Cases. You can then perform similar checks with the *Manage Users* packages in, say, the Use Case and Implementation models.

The **Source** and **Target** field browsers (**[...]**) enable you to examine child packages within the 'theme' package, and obtain further detail on how the feature at this stage is defined.



Other Tools

You can also obtain information on what elements realize and are realized by the elements in a given package, using the Dependency report and Implementation report, respectively (see the *Other Documents* topic in *Report Creation in UML Models*).

You can trace how a Class or Interface element in a diagram or the Project Browser is implemented in code or, for tables, in DDL, using the Source Code viewer. For code, as you click on features in the element, the corresponding code is highlighted in the viewer. (See *Using Enterprise Architect - UML Modeling Tool.*)

From the perspective of model management or project management, you could also use the Audit View 197 as a means of tracing the change history of a package or element. (See *Auditing UML Models.*) The Relationship Matrix also assists in this respect, indicating the impact of changes in one element on others. A Use Case exists because it defines how a Requirement is met; if the Requirement is changed, the Use Case and its dependent diagrams and elements should probably be changed, if not deleted.

17 Reference Data

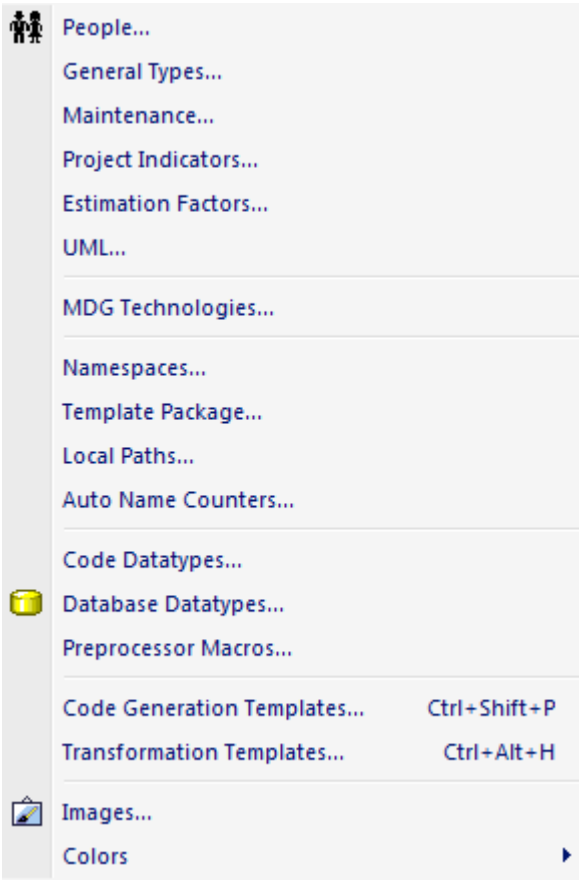
Reference data is used in many places to provide content for drop-down list boxes. Setting up a project often involves setting up the base set of reference types to use. Reference data options can be set up from the **Settings** menu, including:

- People 227
- General Types 235
- Maintenance 242
- Metrics and Estimation 244
- UML 245
- Data Types 249

Having set up the reference data in a project, you can also export and import 250 it between projects.

Note:

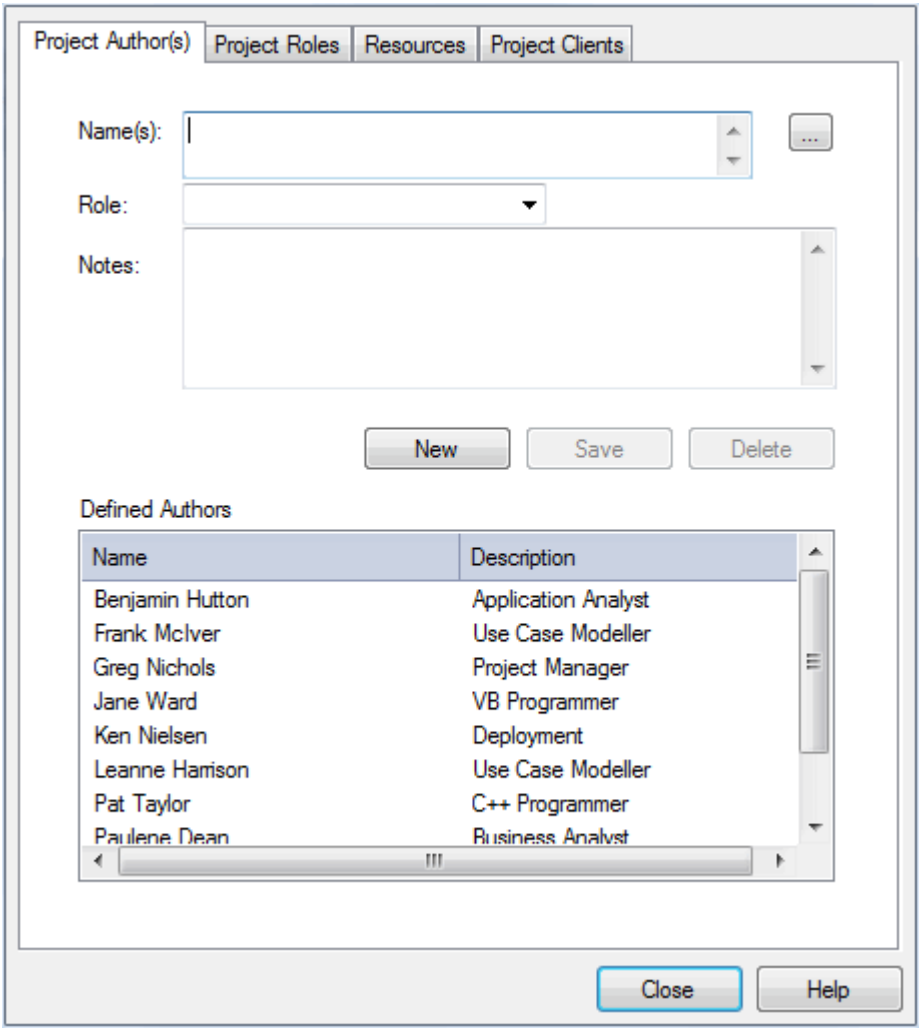
In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Reference Data Update** 183 permission to update and delete reference items. See *User Security in UML Models.*



17.1 People

The People dialog enables you to control the following for your project:

- Project Authors 227
- Project Roles 230
- Project Resources 232
- Project Clients 233

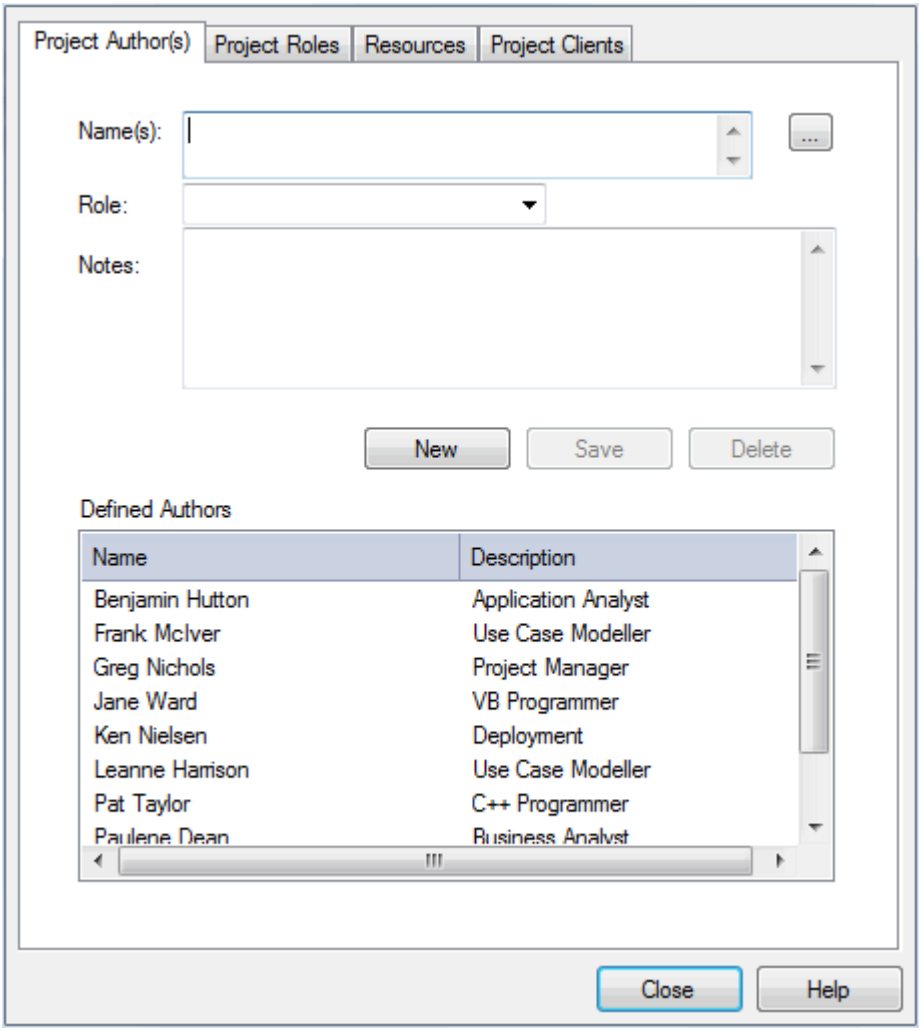


To display this dialog, select the **Settings | People** menu option.

17.1.1 Project Authors

You can define the people who are working on a project, such as the authors of specific elements.

To define the project authors, select the **Settings | People** menu option. The People dialog displays, defaulted to the Project Author(s) tab.



Complete the fields as described below:

Option	Use to
Name	Type the name of the person registered as a Project Author. If you are using a Windows Active Directory, you can select names from the directory. Click on the **[...]** (Browse) button to display the Select Users 229 dialog. You can also type a list of names separated by semi-colons. This enables you to define a group of people sharing a role, such as a team of Developers, Testers or Analysts. Do not leave any spaces between the names and the semicolon. **Note:** If you enter multiple names, Enterprise Architect adds them separately and in alphabetical order to the Defined Authors list. If you then click on one of these names, Enterprise Architect displays that name only in the **Name** field.
Role	Specify the role the Project Author plays in the project (such as **Designer, Analyst,** or **Architect**). You can type a role name or click on the drop-down arrow and select a role defined through the Project Roles 230 tab.

Option	Use to
	Note: If you type a role, this is not added to the roles on the Project Roles tab.
Notes	Type any additional notes concerning the Project Author.
Defined Authors	Review the Project Authors already defined.

Click on the **Save** button to add the new names to the Defined Authors list.

To add further Authors, click on the **New** button.

To delete a Project Author, click on the name in the Defined Authors list and click on the **Delete** button.

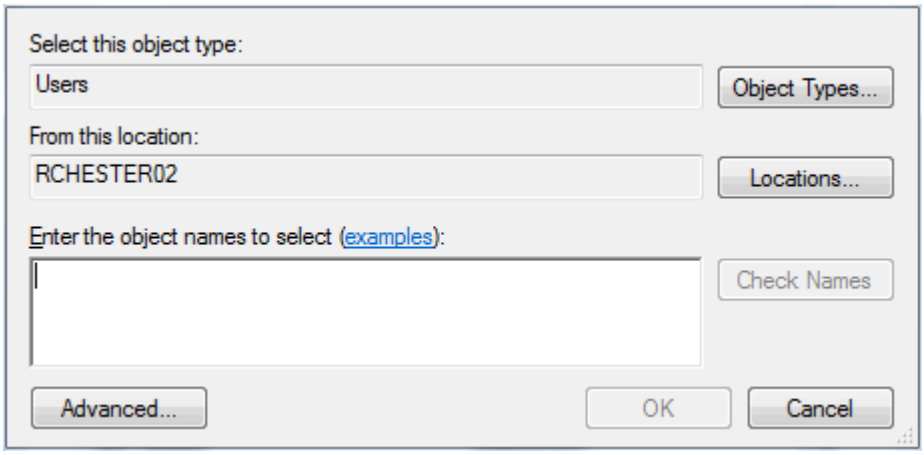
Note:

You can transport these author definitions between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.

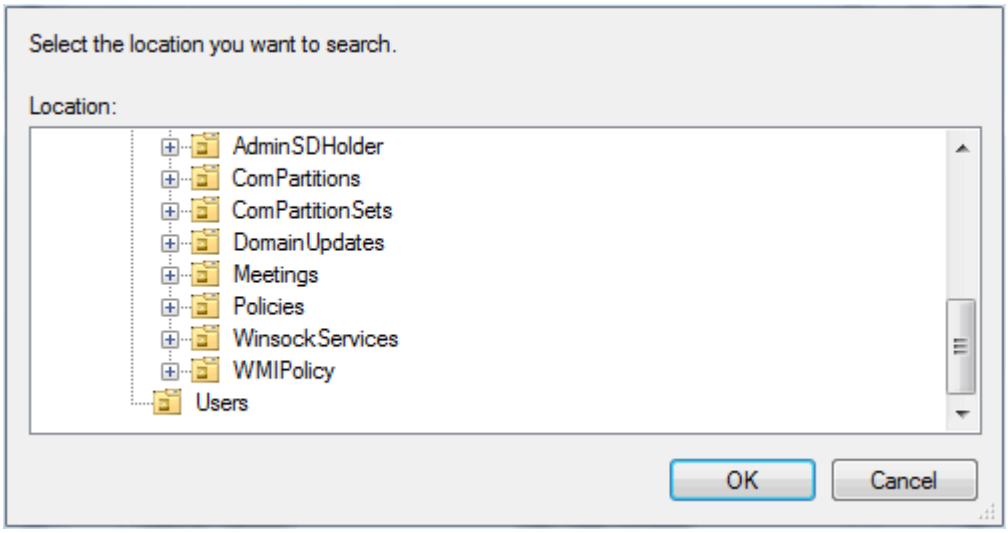
Select from User Directory

If your company is using a Windows Active Directory, you can select the Project Author names from the local or corporate-wide directory. To do this, follow the steps below:

1. On the Project Author(s) tab, click on the **[...]** button. The Select Users dialog displays.



2. Click on the **Object Types** button and select the checkbox for the object type **User**.
3. Click on the **OK** button to return to the Select Users dialog.
4. Click on the **Locations** button. The Locations dialog displays.

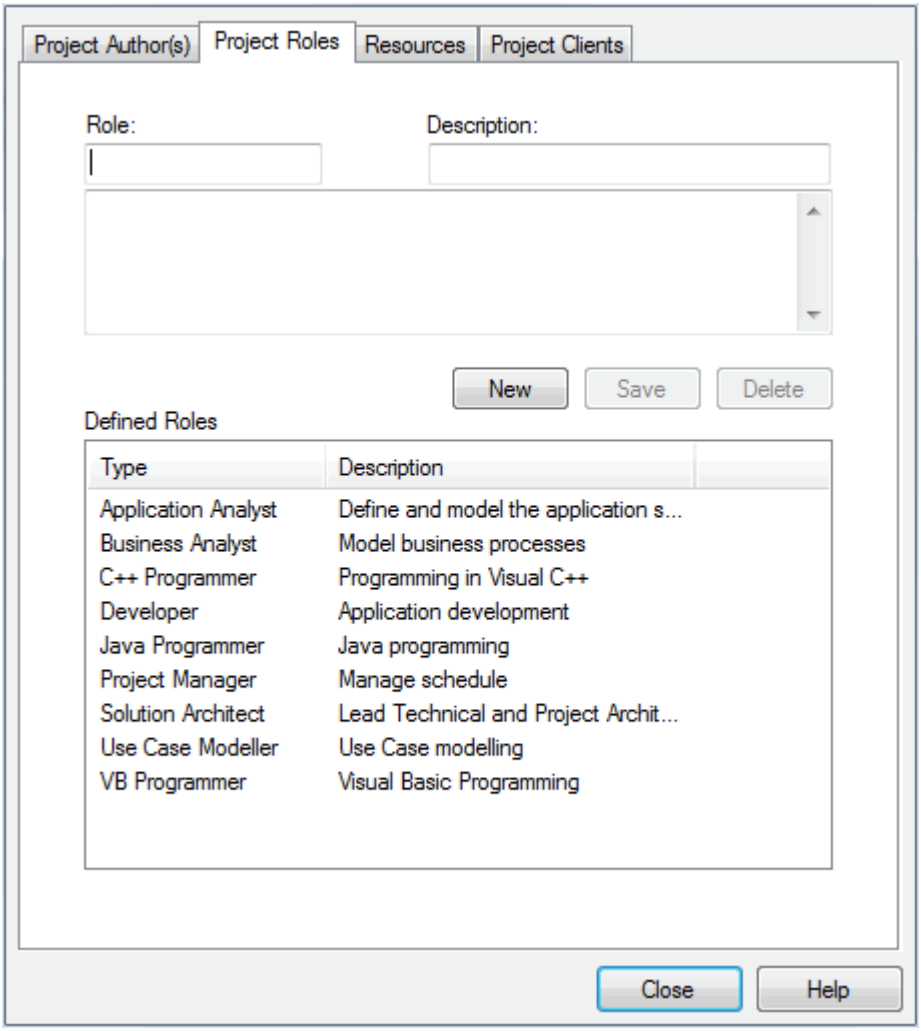


5. Click on the appropriate area or level of the directory, and click on the **OK** button. The Select Users dialog redisplays.
6. In the **Enter the object names to search** field, type the first letter of the user name to search for.
7. Click on the **Check Names** button. The Multiple Names Found dialog displays, listing the names starting with the specified letter found in the directory location.
8. Click on the required name (or press and hold **[Ctrl]** and click on several names), and click on the **OK** button. The simple Select Users dialog redisplays, with the selected names listed.
9. Click on the **OK** button. The Project Authors tab redisplays, with the selected name or names in the **Name(s)** field.

17.1.2 Project Roles

People associated with a project play a *role* in analysis, design or implementation, such as Application Analyst, Architect, Developer and Project Manager. Project roles define the activities that resources can undertake.

To define the role types that are captured within Enterprise Architect, select the **Settings | People** menu option and, on the People dialog, click on the Project Roles tab.



To add further roles, click on the **New** button and complete the fields as described below:

Option	Use to
Role	Type the name of the role.
Description	Type a short description of the role.
Notes	Type any additional information related to the role.
Defined Roles	Review all roles that have been previously defined in Enterprise Architect.

Click on the **Save** button to add the new role to the Defined Roles list.

The Defined Roles list is available for selection for any element in the model; for example, you can select roles on the Project Author 227 tab of the People dialog, and the Resource Allocation tab of the Project Management window. You can also specify other roles on these dialogs, but such roles are not added to the Defined Roles list.

To delete a role, click on the role type in the Defined Roles list and click on the **Delete** button.

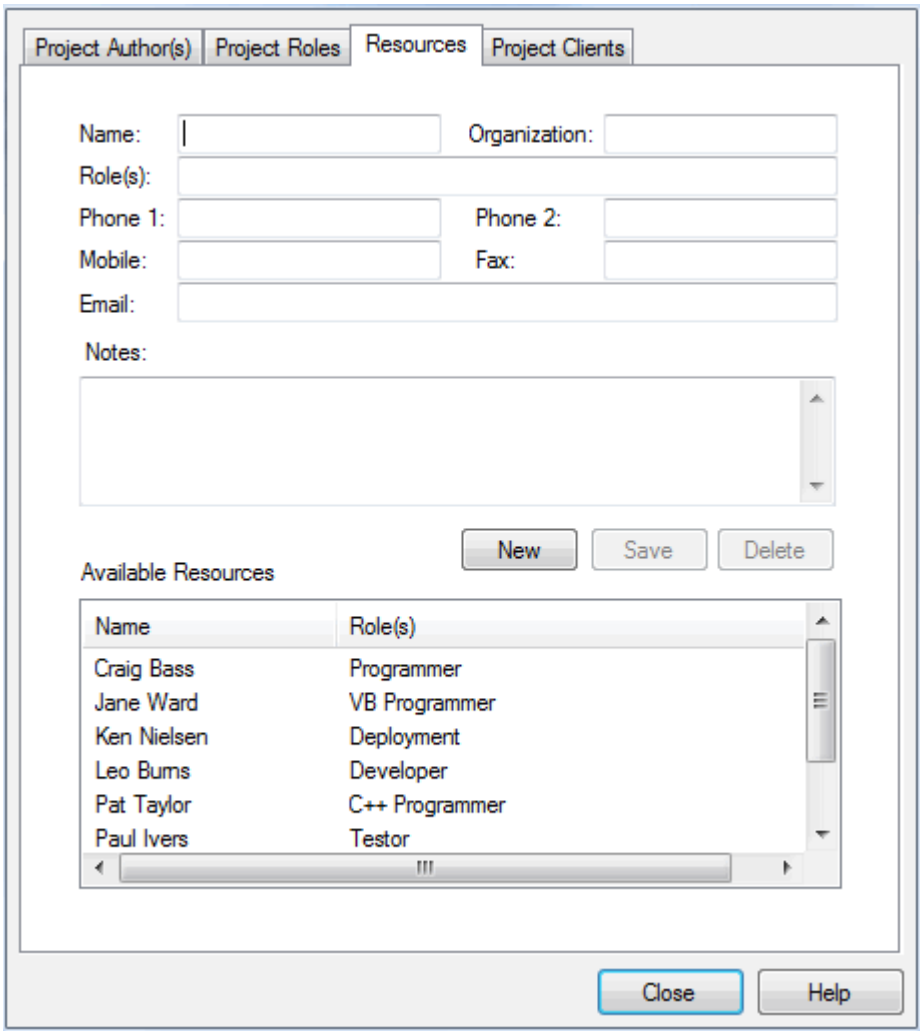
Notes:

- Deleting a role has no effect on any Project Author definition having this role; the deleted role becomes a simple text entry in the Project Author definition.
- You can transport these role definitions between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.

17.1.3 Project Resources

Resources are, for example, project authors, analysts, programmers and architects. That is, anyone who might work on the system over time, either adding to the model or programming and designing elements of the system outside Enterprise Architect.

To record information on project resources, select the **Settings | People** menu option and, on the People dialog, click on the Resources tab.



Complete the fields as described below:

Option	Use to
Name	Type the name of the person listed as a resource. The resource name is available for use in Resource Management. See *Program Management with Enterprise Architect*.

Option	Use to
Organization	Type the name of the organization employing the resource.
Role(s)	Type the role the resource plays in the project (for example, Designer, Analyst, Architect).
Phone 1, Phone 2, Mobile, Fax	Type the contact telephone numbers for the resource.
Email	Type the email address for the resource.
Notes	Type any additional notes on the resource.
Available Resources	Review resources that have already been defined.

Click on the **Save** button to add the new resource to the Available Resources list.

To add further resources, click on the **New** button.

To delete a resource, click on the name in the Available Resources list and click on the **Delete** button.

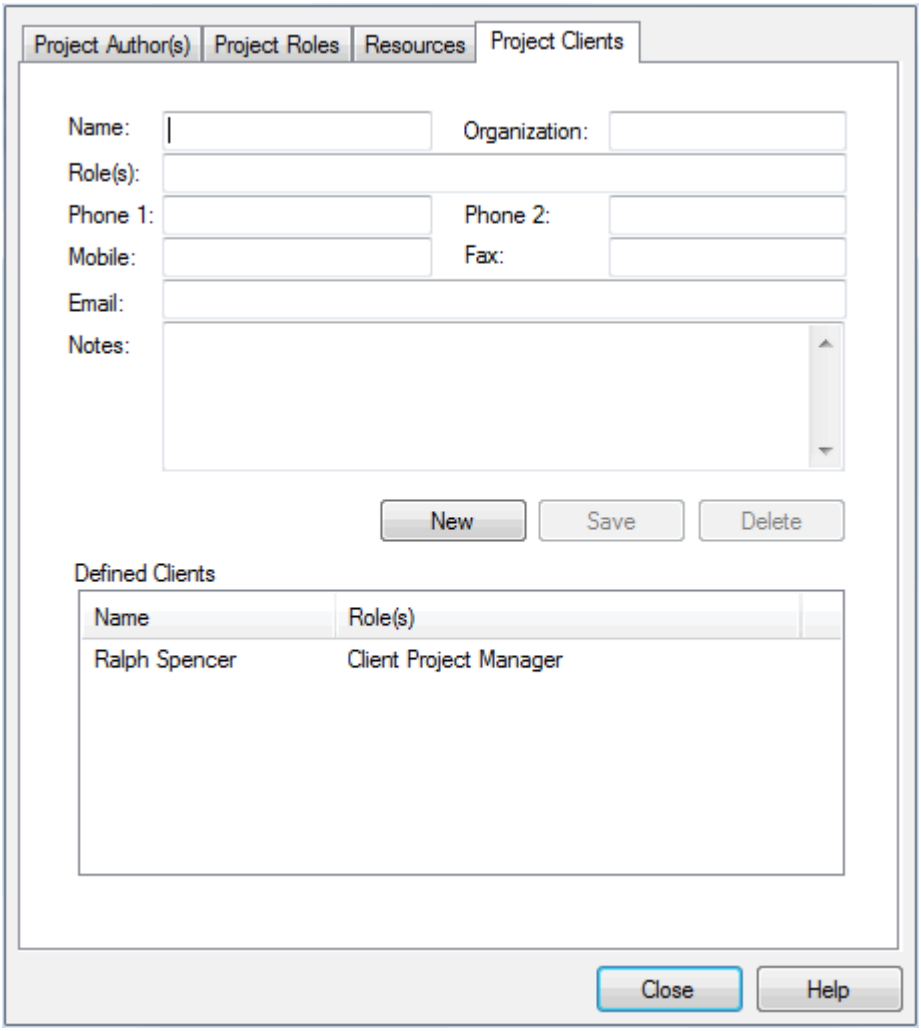
Note:

You can transport these resource definitions between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.

17.1.4 Project Clients

Project clients are the eventual owners of the software system.

To capture client details associated with the current model, select the **Settings | People** menu option and, on the People dialog, click on the Project Clients tab.



Complete the fields as described below:

Option	Use to
Name	Type the name of the client.
Organization	Type the name of the organization that employs the client.
Role(s)	Type the role the client plays in the project (for example, Manager, Sponsor).
Phone 1, Phone 2, Mobile, Fax	Type the contact telephone numbers for the client.
Email	Type the email address of the client.
Notes	Type additional notes on the client.
Defined Clients	Review clients that have already been defined.

Click on the **Save** button to add the new client to the Defined Clients list.

To add details of further clients, click on the **New** button.

To delete a client record, click on the name in the Defined Clients list and click on the **Delete** button.

Note:

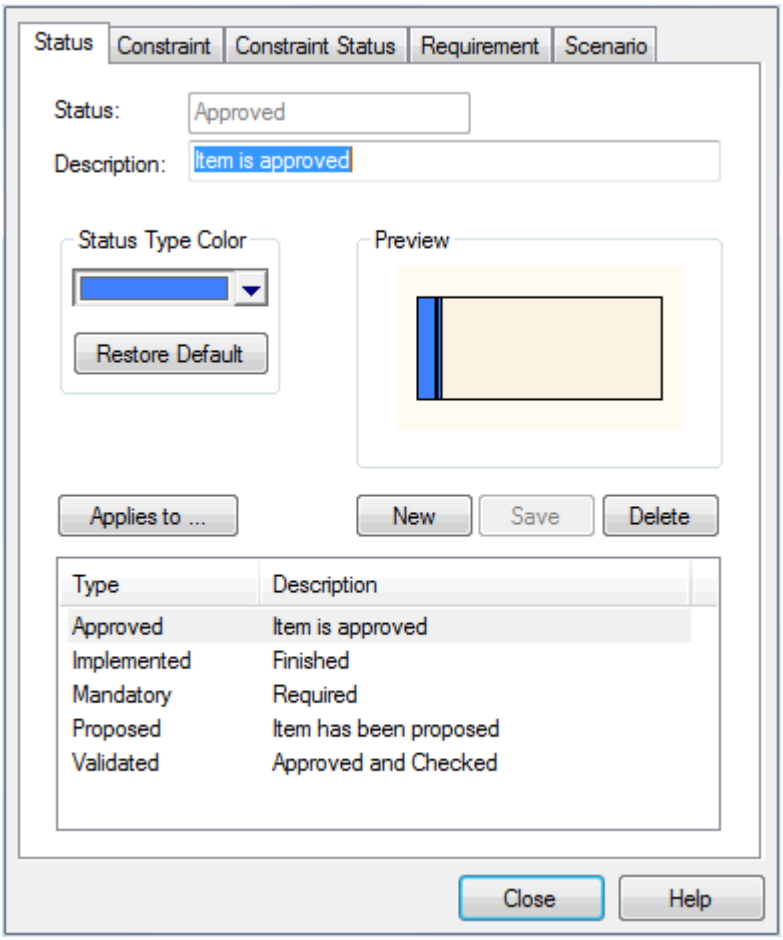
You can transport these client definitions between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.

17.2 General Types

The General Types dialog enables you to configure:

- Status types 235
- Constraint types 237
- Constraint Status types 238
- Requirement types 239
- Scenario types. 240

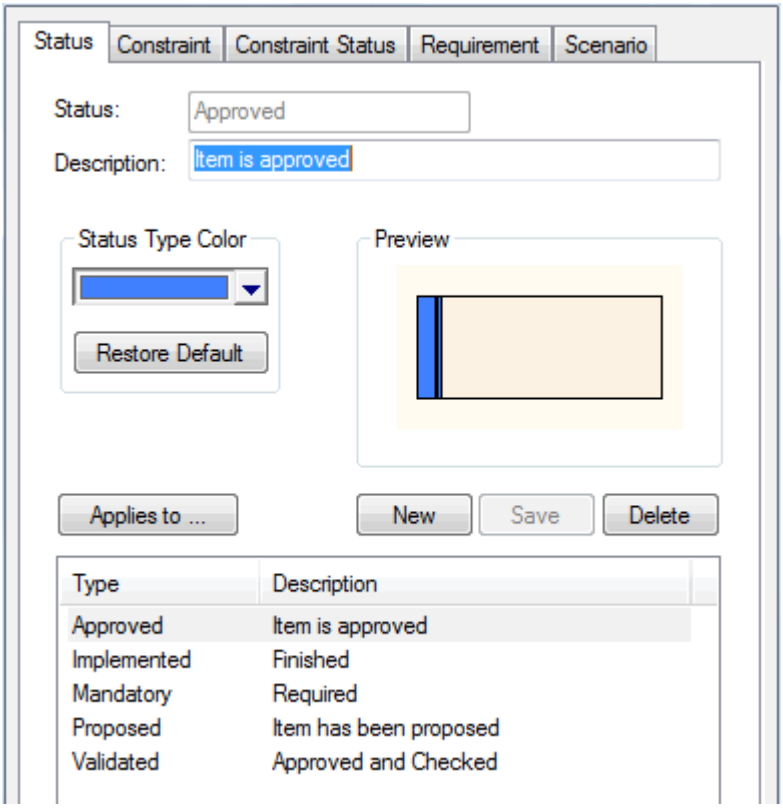
To display the General Types dialog, select the **Settings | General Types** menu option.



17.2.1 Status Types

You can configure a basic list of status types used in Enterprise Architect. Note that whilst most dialogs use this list, not all do so.

To configure status types, select the **Settings | General Types** menu option. The General Types dialog displays at the Status tab.



Create New Status Type

When you display the Status tab, the fields default to the definition of the first type in the Type list. To add a new type, click on the **New** button and:

- In the **Status** field, type the name of the status
- In the **Description** field, type a short description of the status
- Click on the **Save** button.

The status type displays in the Type list. Add the status definition as described in the following sections.

Note:

You can transport the status types (and the colors assigned to status types) between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.

Status Type Color

It is possible to assign a color to each status type, which gives a visual indication of the status of each diagram object. Select a named status type from the Type list, click on the **Status Type Color**drop-down list and select the color for that status. Click on the **Save** button to keep your changes.

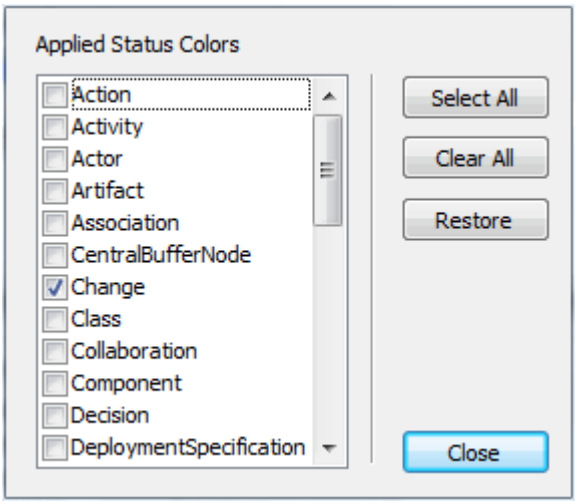
Note:

To ensure status colors display on your diagrams, open the Options dialog at the Objects page and select the **Show status colors on diagrams** checkbox. See the *Defaults and User Settings* topic in *Using Enterprise Architect - UML Modeling Tool.*

Apply Colors to UML Elements

By default, status colors only apply to Requirement, Issue and Change elements (see the *UML Dictionary*). You might decide to also apply these colors to other UML elements, such as Use Cases or Classes. To do this, click on the **Applies to...** button and select the checkbox against each required element type in the

Applied Status Colors list.



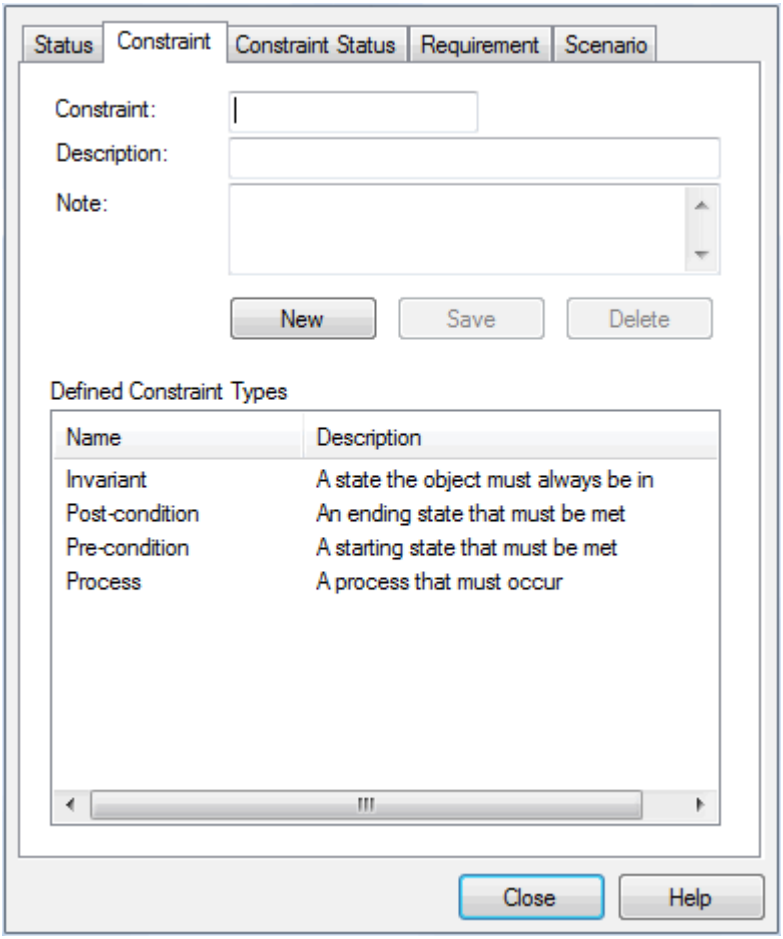
Note:

Requirement, Feature, Issue and Change elements have a status color compartment, but other elements do not. The status color for these elements is applied to the element shadow. Therefore, on the Options dialog Diagram Appearance page you must also select the **Element Shadows on** checkbox. See the *Defaults and User Settings* topic in *Using Enterprise Architect - UML Modeling Tool.*

17.2.2 Constraint Types

The Constraint tab of the General Types dialog enables you to define constraints. These are picked up in a variety of places where constraints might fall into more categories than the basic (default) *Pre-, Post-* and *Invariant* conditions.

To access this dialog, select the **Settings | General Types** menu option. Click on the Constraint tab.



To add a new constraint, click on the **New** button and:

- In the **Constraint** field, type the name of the constraint; for example, *Assumption*
- In the **Description** field, type a brief description of the constraint
- In the **Note** field type any additional information required
- Click on the **Save** button.

The constraint displays in the Defined Constraint Types list.

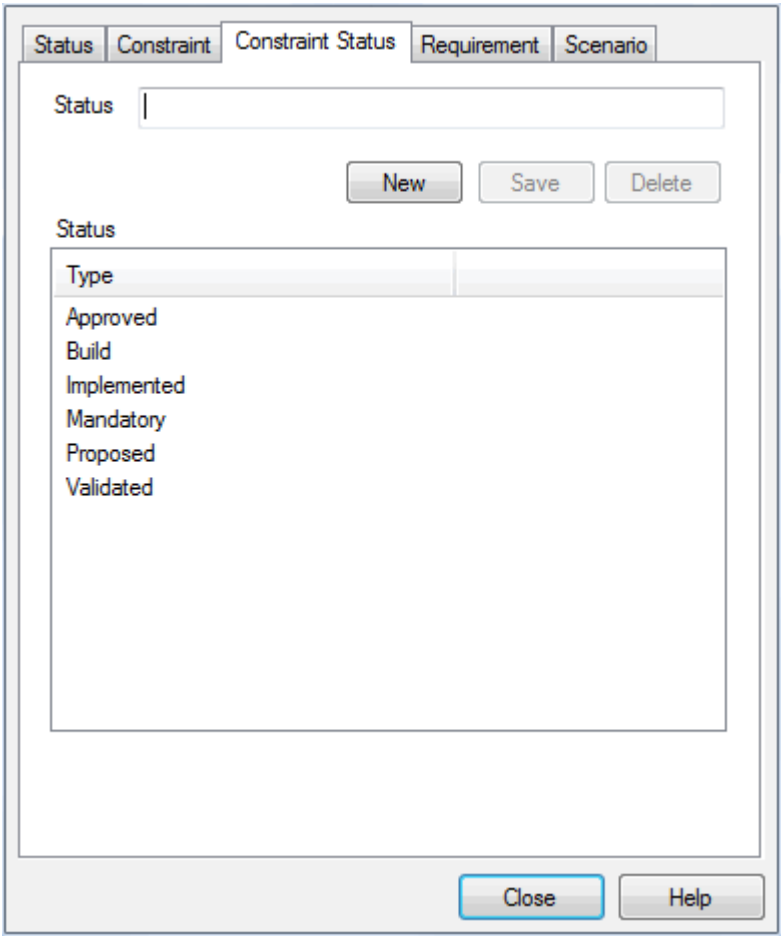
Note:

You can transport these constraints between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.

17.2.3 Constraint Status Types

You can configure the basic list of *constraint status types* used in Enterprise Architect, using the Constraint Status tab of the General Types dialog.

To access this dialog, select the **Settings | General Types** menu option. Click on the Constraint Status tab.



To add a new constraint status type, click on the **New** button, type the status type in the **Status** field, and click on the **Save** button. The constraint status type displays in the Status list.

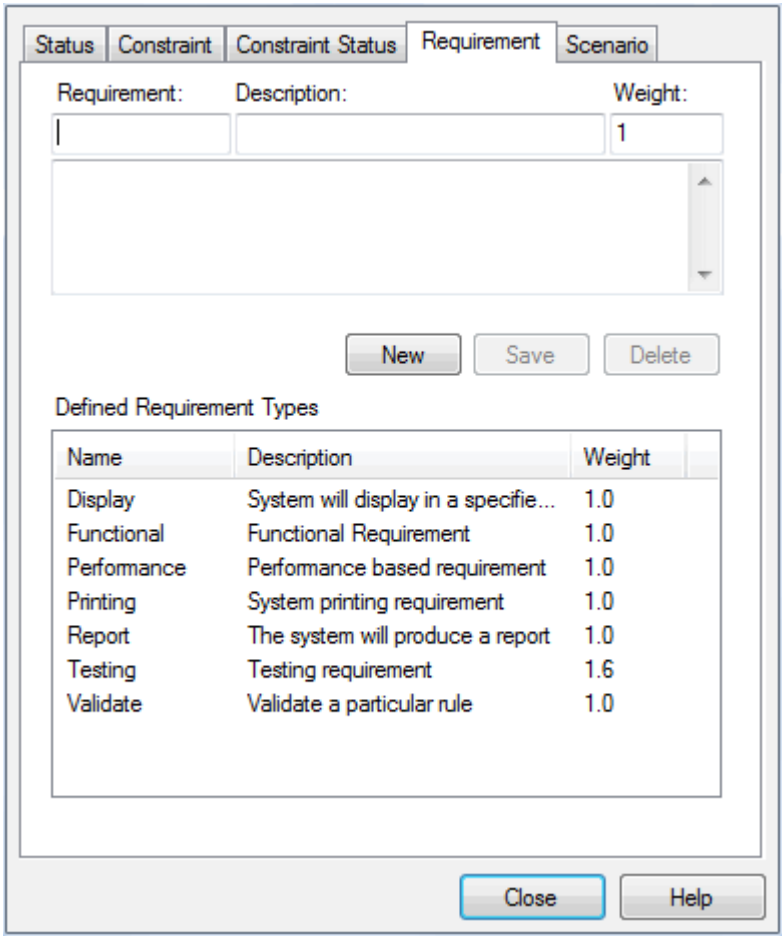
Note:

You can transport these constraint status types between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.

17.2.4 Requirement Types

The Requirement tab of the General Types dialog enables you to specify the generic set of requirement types that can be entered into the requirements sections of dialogs. This helps to maintain a single set of typed requirements.

To access this dialog, select the **Settings | General Types** menu option. Click on the Requirement tab.



To add a new requirement type, click on the **New** button and:

- In the **Requirement** field type the name of the requirement type
- In the **Description** field type a short description of the requirement type
- In the **Weight** field type the weighting to apply to the requirement type
- In the **Note** field, type any additional information on the requirement type
- Click on the **Save** button.

The requirement type displays in the Defined Requirement Types list.

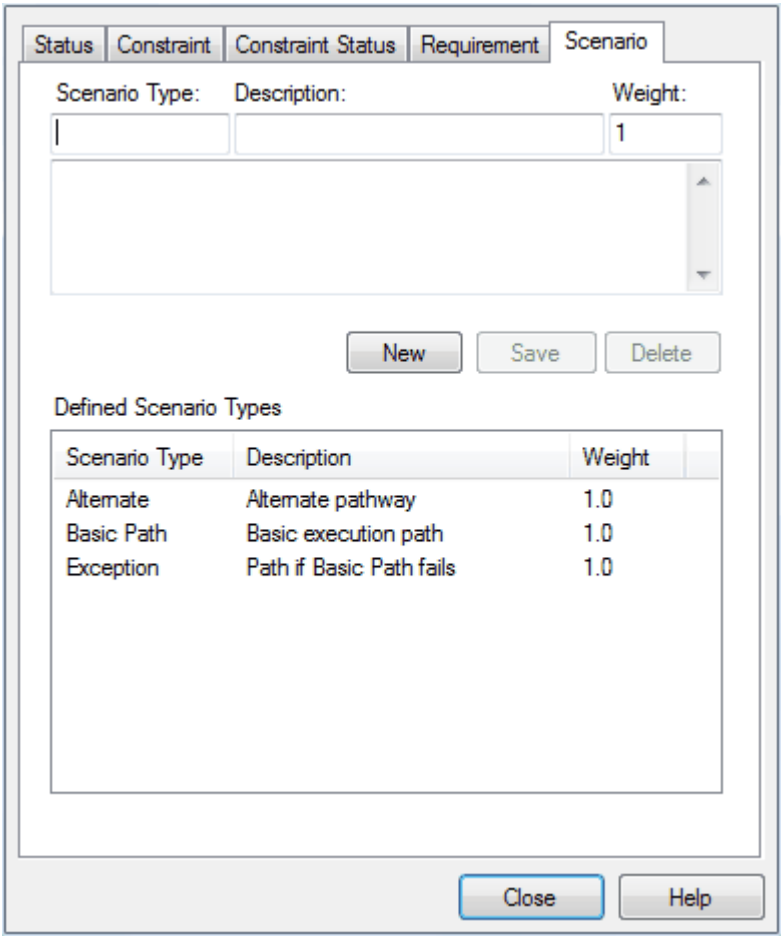
Note:

You can transport these requirement types between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.

17.2.5 Scenario Types

A drop-down list of scenario types is available in the Scenario tab of an element Properties dialog (see the *Working with Elements* section of *UML Modeling With Enterprise Architect - UML Modeling Tool*), with the standard types *Basic Path*, *Exception* and *Alternate Flow*. You can set additional scenario types using the Scenario tab the General Types dialog.

To access this dialog, select the **Settings | General Types** menu option. Click on the Scenario tab.



To add a new scenario type, click on the **New** button and:

- In the **Scenario Type** field type the name of the scenario type
- In the **Description** field type a short description of the scenario type
- In the **Weight** field type the weighting to apply to the scenario type
- In the Note field, type any additional information on the scenario type
- All four fields must be completed. Click on the **Save** button.

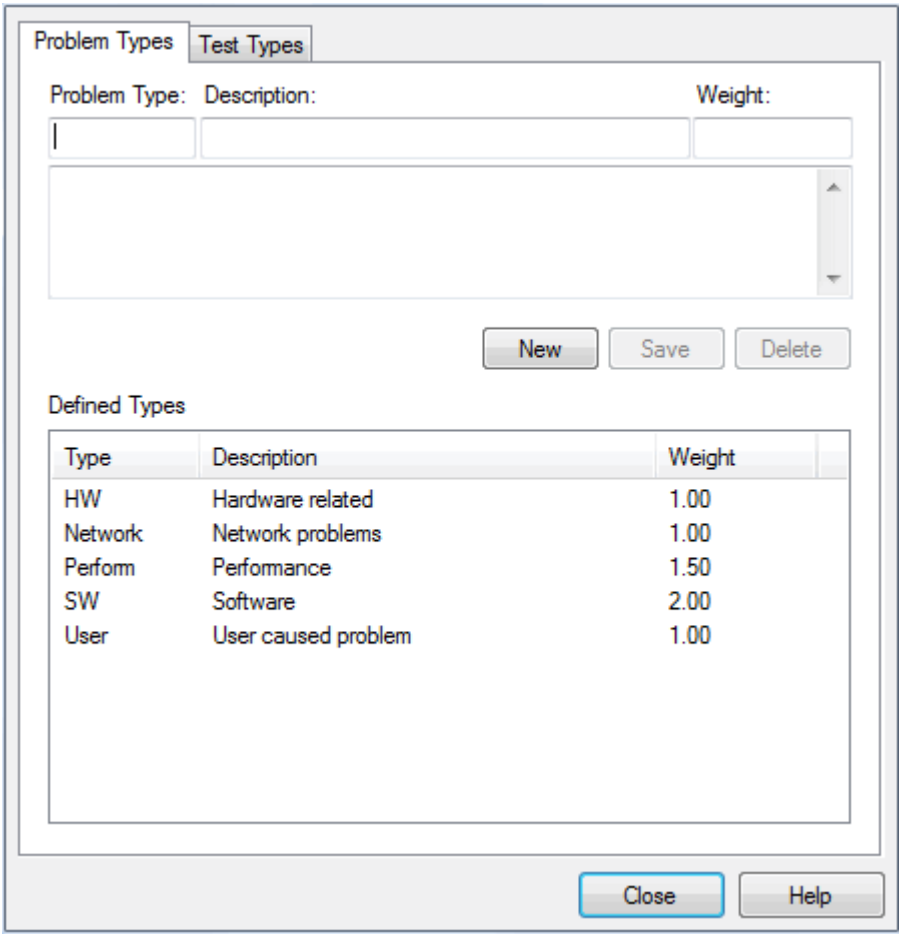
The scenario displays in the Defined Scenario Types list.

Note:

You can transport these scenario types between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.

17.3 Maintenance

To control Testing types 243 for your project, select the **Settings | Maintenance** menu option to display the Maintenance dialog.

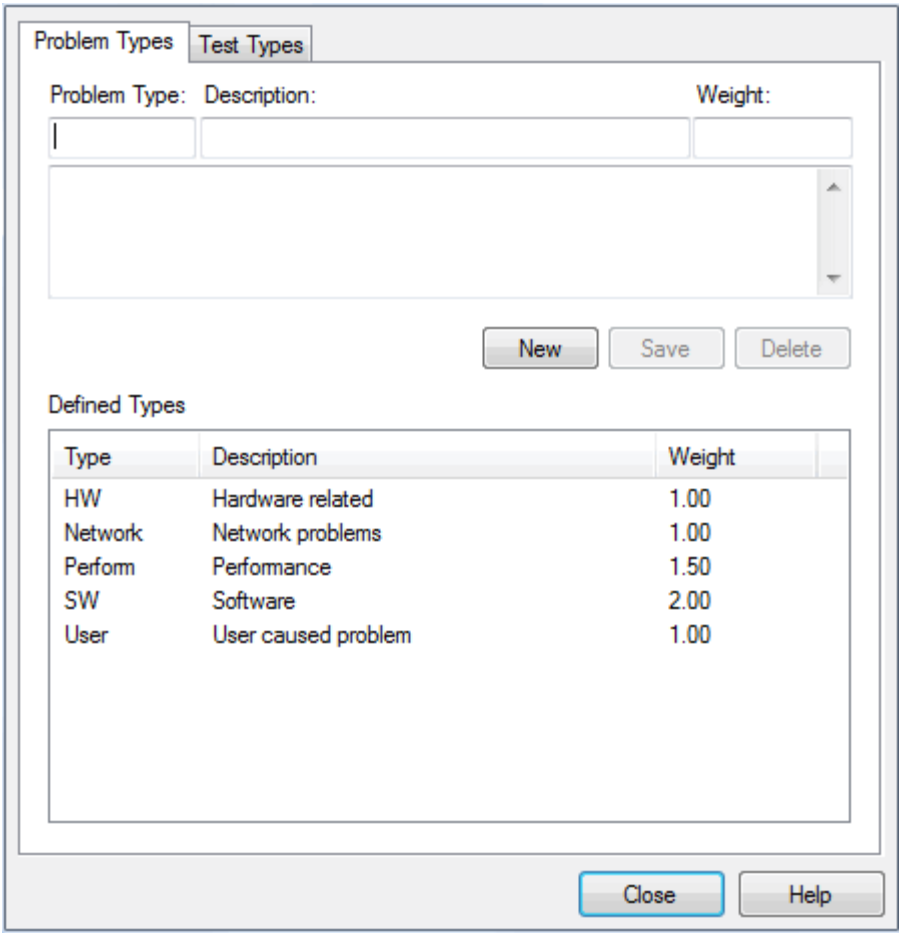


17.3.1 Problem Types

NOT CURRENTLY USED

For the maintenance and change control screens, you can use the Maintenance dialog to set the base *Problem Types* that are handled. Examples are hardware-related issues, performance problems, software bugs and network problems.

To access this dialog, select the **Settings | Maintenance** menu option. The Maintenance dialog displays, defaulting to the Problem Types tab.



To add a new problem type, click on the **New** button and:

- In the **Problem Type** field type the name of the problem type
- In the **Description** field type a short description of the problem type
- In the **Weight** field type the weighting to apply to the problem type
- In the **Note** field, type any additional information on the problem type
- Click on the **Save** button.

The problem type displays in the Defined Types list.

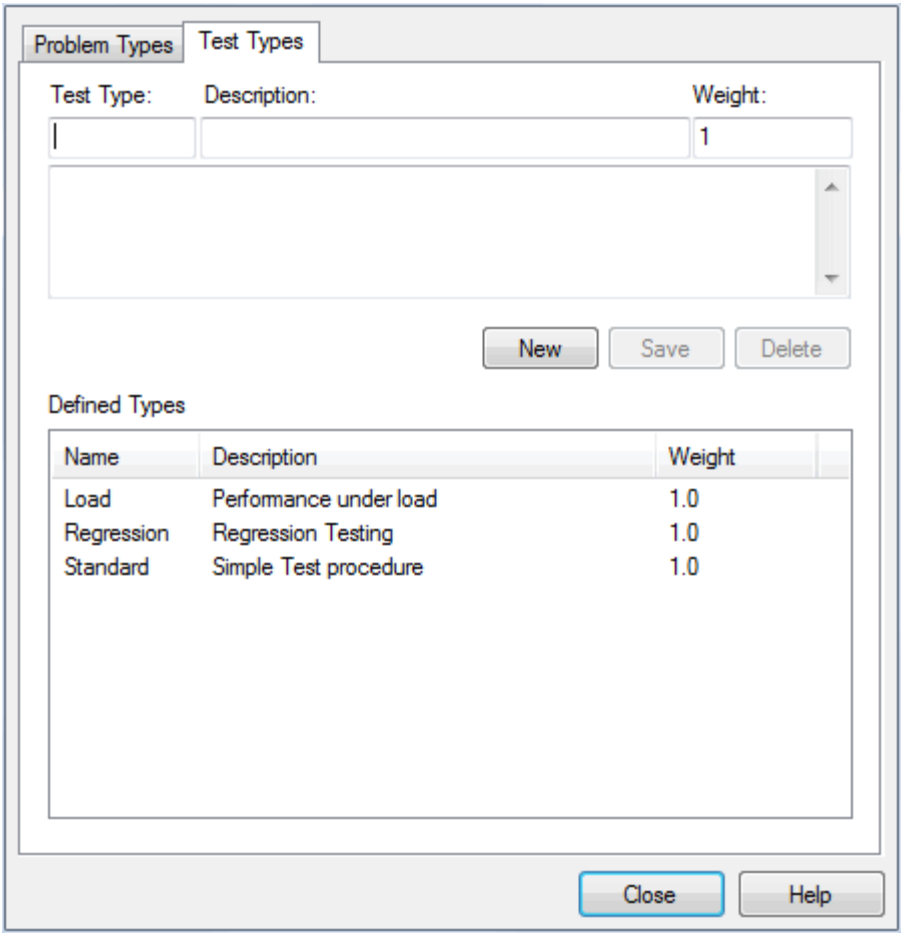
Note:

You can transport these problem types between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu. You transport the problem types together with test types as a *Maintenance Types* file.

17.3.2 Testing Types

Use the Test Types tab of the Maintenance dialog to add testing types to the basic set that comes with Enterprise Architect. Typical test types are load tests, performance tests and function tests.

To access this dialog, select the **Settings | Maintenance** menu option. The Maintenance dialog displays. Click on the Test Types tab.



To add a new test type, click on the **New** button and:

- In the **Test Type** field type the name of the testing type
- In the **Description** field type a short description of the testing type
- In the **Weight** field type the weighting to apply to the testing type
- In the **Note** field, type any additional information on the testing type
- Click on the **Save** button.

The testing type displays in the Defined Types list.

Note:

You can transport these test types between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu. You can either export the test types together with the default problem types, as a *Maintenance Types* file, or separately as a *Test Types* file.

17.4 Metrics and Estimation

TCF values, EFC values and Default Hour Rate for a project are controlled from the Estimation Factors dialog.

Risk, metric and effort types for a project are controlled from the Project Indicators dialog.

For further information on these see the *Project Management* and *Resource Management* topics in *Project Management with Enterprise Architect*, or specifically:

- Technical Complexity Factors
- Environment Complexity Factors
- Default Hours

- Effort Types
- Metric Types
- Risk Types.

17.5 UML Types

The UML Types dialog enables you to configure stereotypes 245, Tagged Value types 247 and the cardinality list 248 for your project.

Select the **Settings | UML** menu option to display this dialog.

17.5.1 Stereotype Settings

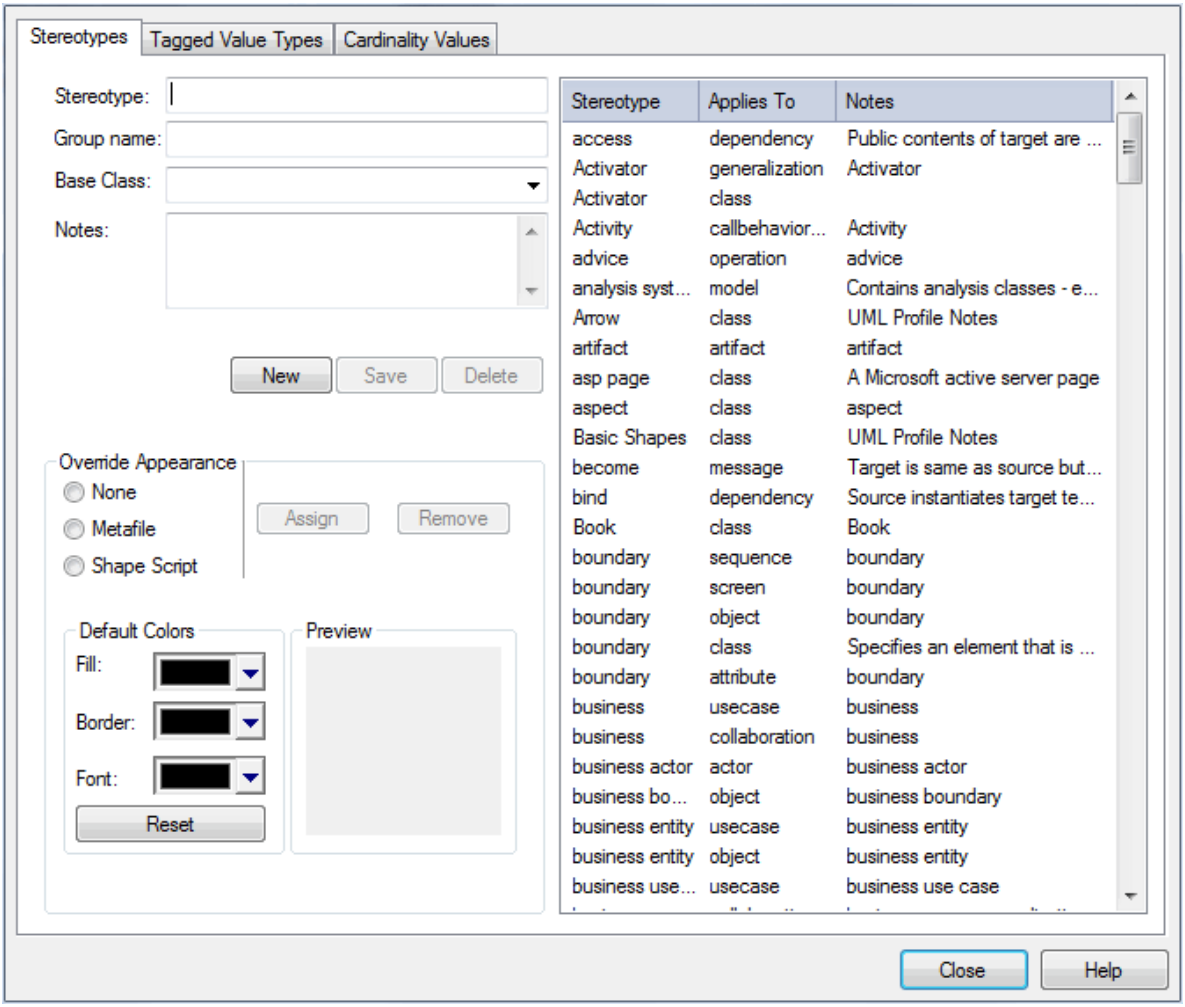
Enterprise Architect has an extensive set of Standard Element Stereotypes that you can apply to any UML construct; see *UML Modeling with Enterprise Architect - UML Modeling Tool*. Using the Stereotypes tab of the UML Types dialog, a Technical Developer can also customize the stereotypes for your project by adding, modifying and deleting them. For information on customizing stereotypes, see *SDK for Enterprise Architect*

Stereotypes can be modified to make use of metafiles (image files) or customized colors, or to make use of the Enterprise Architect Shape Scripts 246 to make new element shapes to determine the shape and dimensions of the element.

Note:

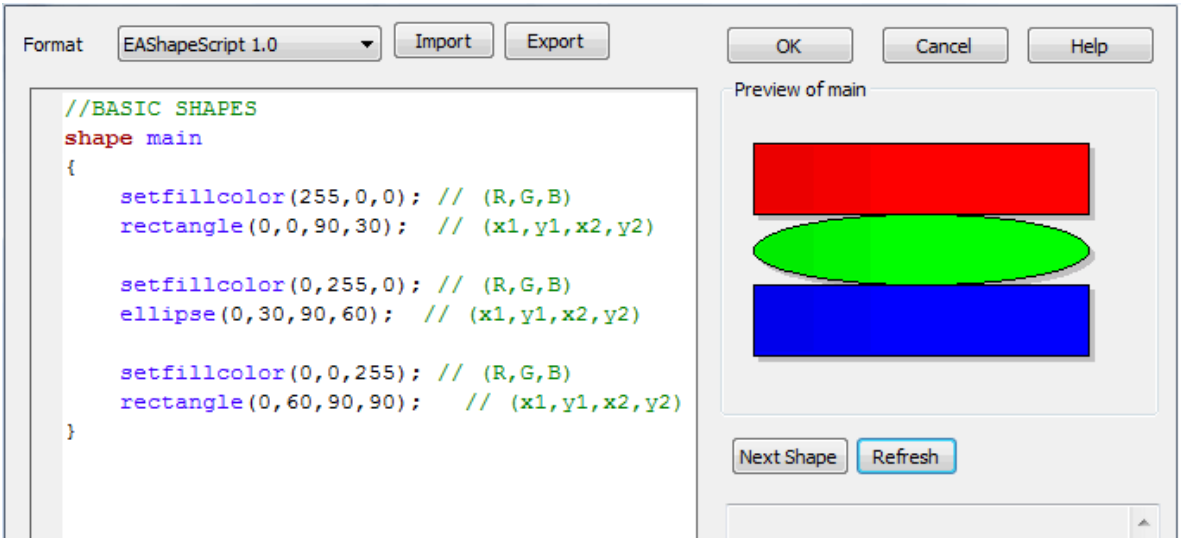
In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Configure Stereotypes** 183 permission to add, modify or delete stereotypes. See *User Security in UML Models.*

To display the Stereotypes tab, select the **Settings | UML** menu option. The UML Types dialog displays, showing the Stereotypes tab.



17.5.1.1 Shape Editor

The Shape Editor enables a Technology Developer to specify custom shapes via a scripting language; that is, to create *Shape Scripts*. These custom shapes are drawn instead of the standard UML notation. Each script is associated with a particular Stereotype, and is drawn for every element of that stereotype.



Notes:

- Shape Scripts adopt the same color gradient settings as normal elements, as defined in the Standard Colors page of the Options dialog.
- If an element's appearance is modified by a Shape Script, many of the **Advanced** context menu options for that element are disabled (see the *Working With Elements* section in *UML Modeling With Enterprise Architect - UML Modeling Tool*).

For information on creating Shape Scripts, see *SDK for Enterprise Architect.*

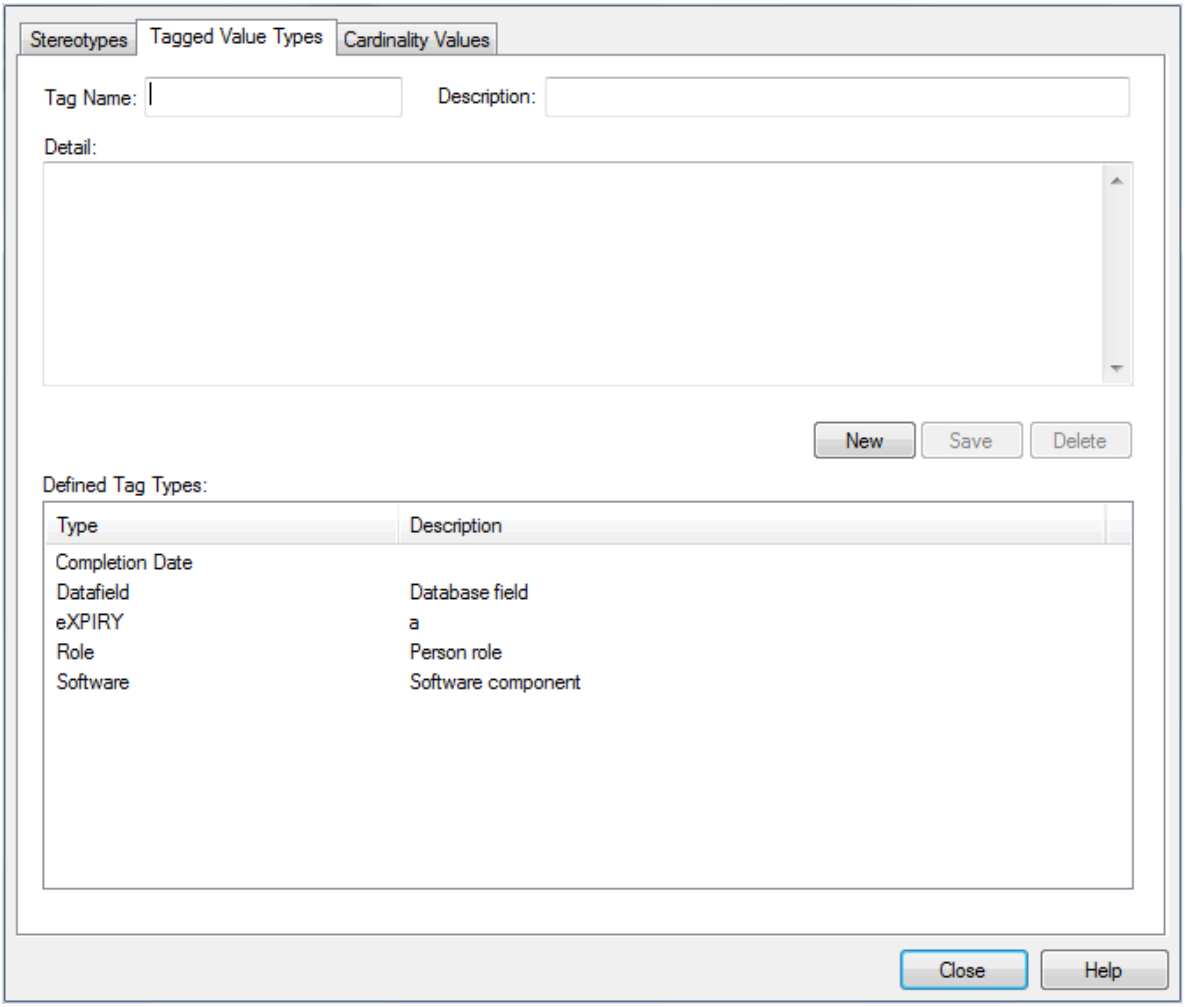
17.5.2 Tagged Value Types

Tagged Values are used in a variety of places within Enterprise Architect to specify additional information about an element or connector. The Tagged Value Types tab of the UML Types dialog enables a Technology Developer to rapidly create Tagged Values, using a range of predefined structured Tagged Values to create structured tags that adhere to a specific format. For example, for model features that use the *predefined* tag *Boolean* you can use the Tagged Values window to assign a value of *True* or *False* and no other value.

You can also add default Tagged Value names and create predefined reference data Tagged Value types and custom masked Tagged Value types.

Any Tagged Value names created display in the drop-down lists of Tagged Value names in the Tagged Value dialogs for elements, operations and attributes. For more information regarding the use of Tagged Values see the *Tagged Values Window* topic in *Using Enterprise Architect - UML Modeling Tool*.

To display the Tagged Value Types tab, select the **Settings | UML** menu option to display the UML Types dialog, and click on the Tagged Value Types tab.



For further information on adding and modifying Tagged Values, see *SDK for Enterprise Architect*.

Note:

You can transport these Tagged Value Type definitions between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu. Tagged Value Types are exported as *Property Types*.

17.5.3 Cardinality

The Cardinality Values tab of the UML Types dialog enables you to add, modify and delete values in the default cardinality list.

The cardinality values are used to define the multiplicity of source and target elements in relationships; see *UML Modeling with Enterprise Architect – UML Modeling Tool*. This is the range of instances of the role that can be active in the relationship; for example, one employee can be assigned to tasks; for the target role you define the range of instances (such as tasks) the employee could be assigned to.

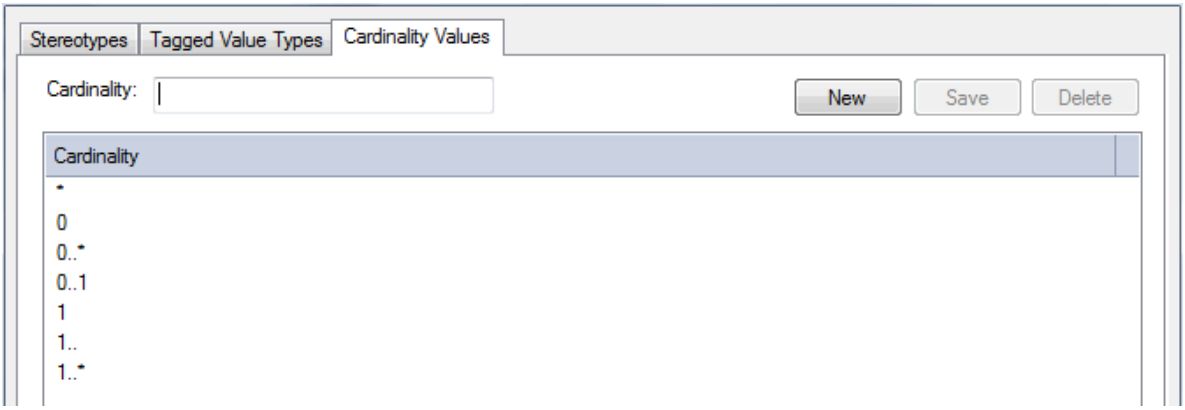
The cardinality values are also used to define the multiplicity of a Classifying element; that is, the number of instances of the element that can exist. For example, the Class element *Building Walls* might have a multiplicity of 2..n, meaning that at least two walls must exist (to support the roof) but there can be many walls if the building design required it.

The values have the following formats:

- *****, or **0..*** - zero, one or many instances
- **0..n** - zero or up to n instances, but no more than n

- **n** - exactly n instances
- **n..*** - n, or more than n instances.

To access this dialog, select the **Settings | UML** menu option. Click on the Cardinality Values tab.



To add a new cardinality value, click on the **New** button. To modify an existing value, click on it in the Cardinality list.

In the **Cardinality** field, type the required cardinality value. Click on the **Save** button.

Note:

You can transport these cardinality values between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.

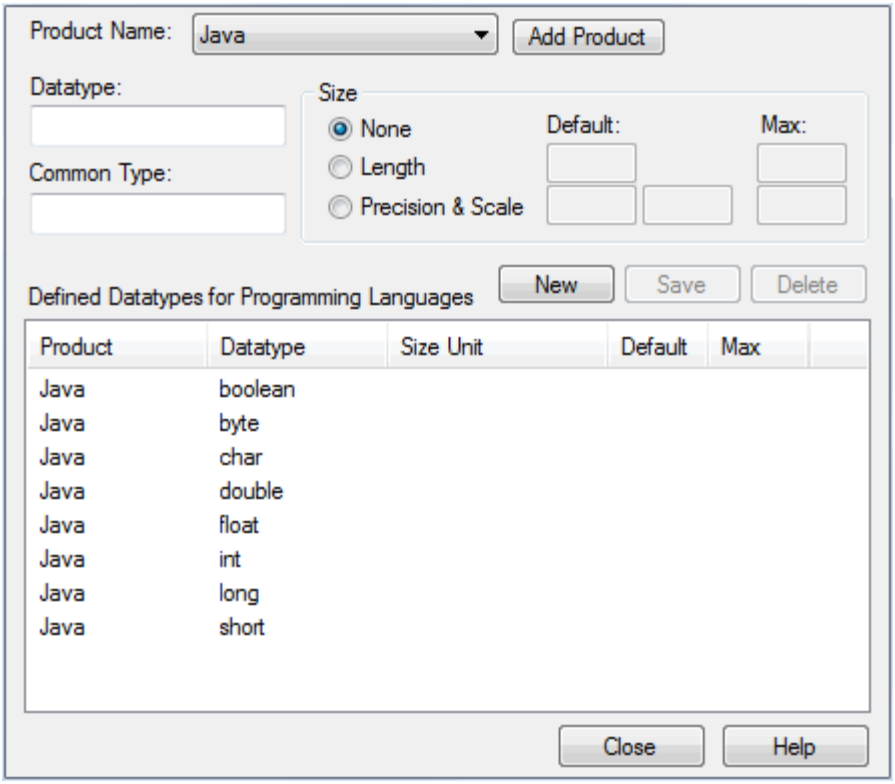
17.6 Data Types

Different programming languages support different inbuilt data types. The Programming Languages Datatypes dialog enables you to extend and manage the set of inbuilt data types associated with a language as well as create new programming languages for use within Enterprise Architect.

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Configure Datatypes** 183 permission to update and delete data types. See *User Security in UML Models.*
- You can delete data types that you have defined, but you cannot delete any of the predefined data types.

To access this dialog, select the **Settings | Code Datatypes** menu option.



Option	Use to
Product Name	Specify the name of the programming language.
Add Product	Add a new programming language to the drop-down fields for Class elements within the Enterprise Architect model and enable the new language to be made available to the Code Template Editor once at least one datatype has been added to the language. See *Code Engineering Using UML Models.*
Datatype	Specify the name of the datatype; this is the language-specific name of the datatype.
Common Type	Specify the common type, the generic name of the datatype; for example, the Java *boolean* datatype has a common datatype *Boolean*.
New	Create a new data type.
Save	Save the newly created datatype.
Delete	Delete the selected datatype.

Note:

You can transport these data types between models, using the **Export Reference Data** 251 and **Import Reference Data** 253 options on the **Tools** menu.

17.7 Import and Export Reference Data

Reference data (including Glossary and Issue information) can be exported to and imported from .XML files for convenient update of models.

You can automatically 253 or manually 253 import data into the model from a reference data .XML file, exported 251 from another model or an iteration of the current model. The automatic import is conditional on there

being changes to the source file since the last import, but you can also configure Enterprise Architect to display a prompt for you to allow or cancel the import.

Examples of where exporting and importing reference data can be useful include:

- Copying glossaries from one model to another
- Adding additional stereotype profiles by merging new stereotypes into the model
- Updating reference data from files supplied by Sparx Systems as a maintenance release
- Copying resources, clients and so on from one model to another.

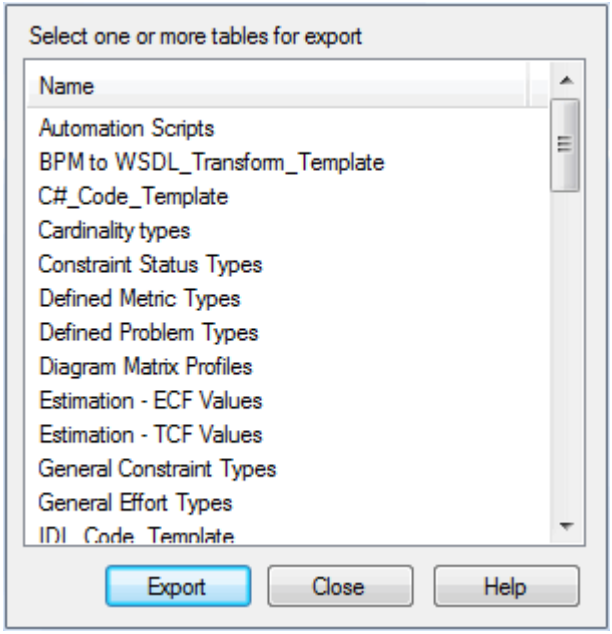
17.7.1 Export Reference Data

When reference and project data is exported, Enterprise Architect writes it out to a custom XML file. This includes table information, filter information, rows and columns.

Procedure

To export data, follow the steps below:

1. Select the **Tools | Export Reference Data** menu option. The Data Exporter dialog displays.



2. From the Name list, select the table to export. You can select one or more tables for a single file.
3. Click on the **Export** button.
4. When prompted to do so, enter a valid file name with a .XML extension.

This exports the data to the file. You can use any text or XML viewer to examine this file.

The data exported includes all instances of the data type in the project; for example, all defined cardinality values, or all RTF Style Templates.

Note:

If there are no instances of a selected data type in the project, the export does not generate any output for that data type in the XML file.

For information on each category of data you can export, refer to the following topics:

- ActionScript Code Templates (and other language code templates; see the *Code Templates* topic in *Code Engineering Using UML Models*); entries for code templates are listed only if templates for a particular language exist in the model. The list entry has the format <language_name>_Code_Templates.
- Automation Scripts (JavaScript, JScript and VBScript)

- Cardinality Types 248
- Constraint Status Types 238
- CSV 118 Specifications
- Defined Metric Types (see *Project Management with Enterprise Architect)*
- Defined Problem Types 242
- Diagram Matrix Profiles (Model Profiles) (see the *Swimlanes Matrix* topic in *UML Modeling with Enterprise Architect – UML Modeling Tool*)
- Estimation - Environment Complexity Factor Values (see *Project Management with Enterprise Architect*)
- Estimation - Technical Complexity Factor Values (see *Project Management with Enterprise Architect*)
- General Constraint Types 237
- General Effort Types (see *Project Management with Enterprise Architect*)
- Import Component Types (see the *Code Engineering Settings* topic in *Code Engineering Using UML Models*)
- Linked Document Templates (see the *Create Linked Document Templates* topic in *UML Modeling with Enterprise Architect – UML Modeling Tool*)
- Macro List (Preprocessor macros) (see the *Language Macros* topic in *Code Engineering Using UML Models*)
- Maintenance Types 242 (both Problem Types and Test Types)
- Model Authors 227
- Model Data Types - Code 249 and DDL (see the *Generate DDL* topic in *Code Engineering Using UML Models*)
- Model Images (see the *Using the Image Manager* topic in *UML Modeling with Enterprise Architect – UML Modeling Tool*)
- Project Clients 233
- Project Glossary (see *Project Management with Enterprise Architect*)
- Project Issues (see *Project Management with Enterprise Architect*)
- Project Resources 232
- Project Roles 230
- Project Tasks (see *Project Management with Enterprise Architect*)
- Property Types (Tagged Value Types) (see *SDK for Enterprise Architect*)
- Requirement Types 239
- Risk Types (see *Project Management with Enterprise Architect*)
- Scenario Types 240
- Security - Group Permission 182 (see *User Security in UML Models*)
- Security - Groups 182 (see *User Security in UML Models*)
- Security User Groups 179 (see *User Security in UML Models*)
- Security - User Permissions 180 (see *User Security in UML Models*)
- Security - Users 175 (see *User Security in UML Models*)
- Standard Complexity Types - currently, these cannot be directly edited and are therefore effectively standard for all models; they can be listed using the Predefined Reference Data Tagged Value type *ComplexityTypes*. (see *SDK for Enterprise Architect*)
- Status Colors 235 (the colors defined for status types)
- Status Types 235
- Status - Applies To 235 (elements to which the status can be applied)
- Stereotypes (all those listed on the Stereotypes page of the UML Types dialog) (see *SDK for Enterprise Architect*)
- Templates - HTML Style (exports the web templates listed in the *Templates* folder of the Resources window) (see *Report Creation in UML Models*)
- Templates - HTML Style Detail (exports the content of the HTML report templates) (see *Report Creation in UML Models*)
- Templates - RTF Document (exports the Extended RTF Style templates in the *Templates* folder of the Resources window) (see *Report Creation in UML Models*)
- Templates - RTF Style (exports the Legacy RTF Style templates in the *Templates* folder of the Resources window) (see *Report Creation in UML Models*)
- Templates - RTF Style Detail (exports the content of the RTF templates) (see *Report Creation in UML Models*)
- Templates - RTF Tag and Language Options (exports RTF word substitution templates) (see *Report*

Creation in UML Models)

- Test Types 243

17.7.2 Import Reference Data

It is possible to import reference data into your model in Enterprise Architect, from a reference data .xml file exported from another model or an iteration of the current model, either:

- Manually 253, as required, whenever you know there is new or changed data to apply, or
- Automatically 253 whenever the model is reloaded into Enterprise Architect (if the file has changed since the previous import).

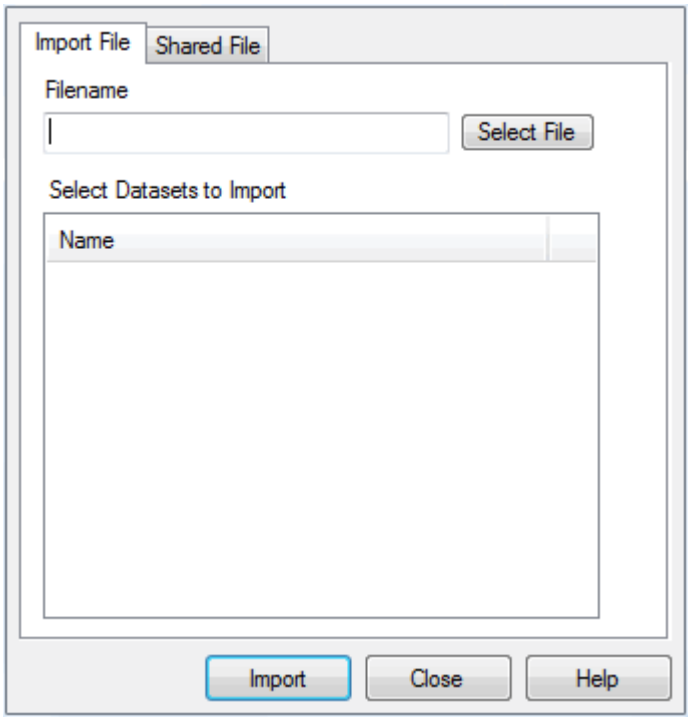
When you import data into Enterprise Architect, the system merges the incoming data with the existing data. If a record already exists it is updated to the new values. If the record does not exist, Enterprise Architect adds a new record. Enterprise Architect never deletes records.

For a list and explanation of the data types you could import, see the *Export Reference Data* 251 topic.

Import Data Manually

To import data manually, follow the steps below:

1. Select the **Tools | Import Reference Data** menu option. The Import Reference Data dialog displays.



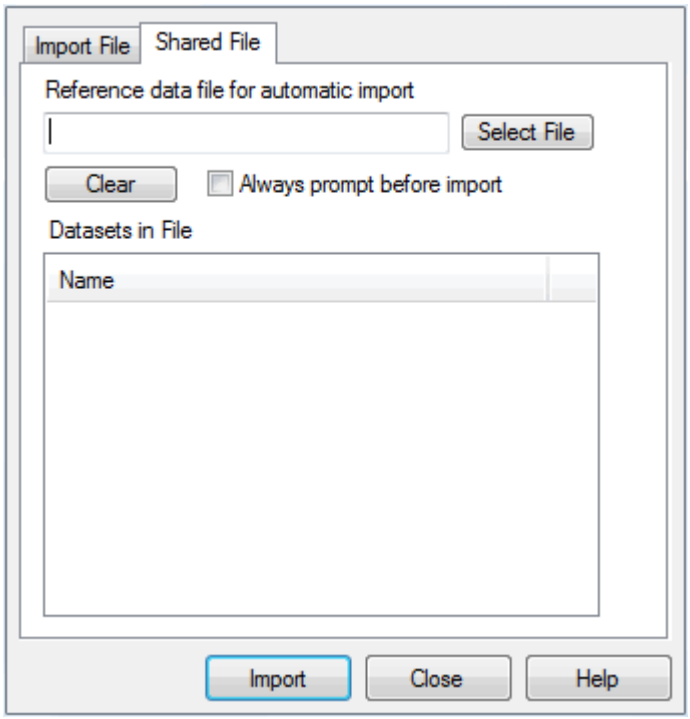
2. Click on the Import File tab and on the **Select File** button, then select the filename to import data from. This must be an XML file produced by the Enterprise Architect Data Exporter 251.
3. If you have entered the name of a valid file, a list of available tables to import displays in the Select Datasets to Import panel.
4. Click on one or more of the tables to import (press **[Ctrl]** or **[Shift]** to click on multiple tables).
5. Click on the **Import** button to start the process. A message displays when the import is complete. Generally the process is quite fast.

Import Data Automatically

The automatic import checks if the source file has changed since the last import; if the file has not changed, the import does not proceed. If the file has changed the changed data is imported; however, you can configure Enterprise Architect to display a prompt for you to allow or cancel the import.

To set up the system to check and import reference data automatically whenever your model is reloaded into Enterprise Architect, follow the steps below:

1. Select the **Tools | Import Reference Data** menu option. The Import Reference Data dialog displays. Click on the Shared File tab.



2. If you are changing an existing configuration to import from a different .XML file, click on the **Clear** button to clear the dialog fields.
3. Click on the **Select File** button and browse for the filename to import data from. This must be an XML file produced by the Enterprise Architect Data Exporter 251.
4. If you have entered the name of a valid file, a list of tables to import displays in the Datasets in File panel.
5. If you prefer to control whether or not the automatic import takes place, select the **Always prompt before import** checkbox.
6. Click on the **Import** button to import the reference data now, and to enable the automatic check and import for subsequent reloads.

Index

- . -

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- I -

- L -

- M -

- N -

- O -

- P -

- Q -

- R -

- S -

- T -

- U -

- V -

- W -

- X -

UML Model Management

www.sparxsystems.com