

# An Introduction to Enterprise Architect

# Requirements Management with Enterprise Architect

By Sparx Systems

*All material © Sparx Systems 2010 – version 1.3*

[www.sparxsystems.com](http://www.sparxsystems.com)

## Trademarks

Object Management Group, OMG, Unified Modeling Language, UML, are registered trademarks or trademarks of the Object Management Group, Inc.

Microsoft, MS Word and Excel are trademarks or registered trademarks of Microsoft Corporation.

All other product and / or company names mentioned within this document are used for identification purposes only, and may be trademarks or registered trademarks of their respective owners.

## Table of Contents

<b>INTRODUCTION</b> .....	<b>4</b>
<b>REQUIREMENTS MANAGEMENT WITH UML</b> .....	<b>4</b>
GLOSSARY OF TERMS .....	4
<b>GETTING STARTED WITH REQUIREMENTS MANAGEMENT</b> .....	<b>5</b>
<b>REQUIREMENTS MODELING</b> .....	<b>5</b>
EXTERNAL REQUIREMENT ATTRIBUTES .....	6
<i>Useful Features for External Requirements</i> .....	7
Adding Additional Attributes to Requirements.....	8
Predefining Tagged Values Types for Requirements.....	8
Element Numbering.....	10
Different Views of Requirements Using the Element List View .....	12
TRACEABILITY AND RELATING EXTERNAL REQUIREMENTS .....	15
<i>Creating Relationships Using Diagrams</i> .....	15
Creating a Common Diagram .....	15
Types of Relationships .....	16
The Relationship Matrix.....	17
Using the Traceability Window .....	18
CHANGE CONTROL .....	19
Auditing .....	19
Using Baselines .....	21
Change Requests and Issues on External Requirements .....	21
Using the Maintenance View .....	21
Using Maintenance Elements for Changes and Issues .....	22
INTERNAL REQUIREMENTS .....	23
<b>CREATING CUSTOMER QUALITY REQUIREMENTS DOCUMENTATION</b> .....	<b>25</b>
EXTERNAL REQUIREMENTS REPORTS .....	25
INTERNAL REQUIREMENTS REPORTS .....	26
IMPLEMENTATION REPORT, DEPENDENCY REPORT, AND THE ELEMENT LIST VIEW .....	27
Implementation Report .....	27
Dependency Report.....	27
Element List View .....	27
<b>ADDITIONAL REQUIREMENTS MANAGEMENT FEATURES</b> .....	<b>28</b>
CREATING YOUR OWN REQUIREMENT TYPES .....	28
COLOR CODING REQUIREMENTS.....	29
AUTO ELEMENT NAMING .....	29
DRAG AND DROP REALIZATIONS .....	30
IMPORTING EXTERNAL REQUIREMENTS USING THE CSV IMPORT .....	30
Import a Hierarchy of Requirements .....	34
DOCUMENT BASED REQUIREMENT CREATION.....	36
<b>AN INTRODUCTION TO USE CASES IN ENTERPRISE ARCHITECT</b> .....	<b>38</b>
USE CASE DIAGRAMS .....	38
DEFINING SCENARIOS .....	39
LINKING WITH REQUIREMENTS.....	40
<b>ADDITIONAL FEATURES OF ENTERPRISE ARCHITECT</b> .....	<b>40</b>
ATTACHING DOCUMENTS AND FILES .....	40
THE GLOSSARY FUNCTION .....	42
DEFINING REQUIREMENT ATTRIBUTES USING A PROFILE OR A TEMPLATE.....	44

## Introduction

Enterprise Architect (EA) is one of the few UML tools that integrate requirements management with other software development disciplines, by creating requirements directly in the model. Requirements management is built into the core product, solving many of the issues of traceability, interdisciplinary team divisions, integration with change and configuration management systems. Some of the features include:

- The ability to create and view requirements directly in the model.
- The ability to detail use cases directly in the model.
- The ability to enter attributes for each requirement such as difficulty, status, and type; you can also define your own attributes.
- The ability to trace requirements to business rules, test cases and analysis artifacts.
- A Relationship Matrix for traceability and viewing the impact of changes to requirements.
- The ability to create customer quality MS Word and HTML documentation.

## Requirements Management with UML

The management of requirements has traditionally been one of the more difficult and problematic disciplines in the software development industry. There are a number of reasons for this, but perhaps the most significant are the following:

- Diverse group input into the requirements
- Organizational boundary divisions
- Tool boundary divisions
- Volatility of requirements
- Imprecision and ambiguities of natural languages

The UML and Enterprise Architect can be used to reduce (and in many circumstances remove) these problems. The UML has introduced a new way to describe functional requirements, the use case. While this is a welcome addition to the requirements analyst's toolbox, the lack of clear guidelines about their application has led to some misconceptions, and a myriad different use case styles and interpretations. The introduction of the UML use case has also led to confusion as to where to place other requirements, such as security requirements, performance requirements and quality ranges. In this paper we will discuss many of these issues and how to use Enterprise Architect to create and manage requirements effectively.

## Glossary of Terms

There are several terms used in this document which you may not be familiar with. The following is a list of terms, and how they relate to requirements management and Enterprise Architect.

- Element – A generic term referring to a singular object in a model. Some of the common elements you will come across include requirements, actors and systems.
- External requirement – A requirement that is modeled as an element.
- Internal requirement – A requirement that is modeled as the “responsibility” of an existing element.

- Model – A representation of a particular system, such as a business process or a database.
- Diagram – A common way of representing the way in which models and elements interact. The currently open diagram is usually located in the centre of the Enterprise Architect interface.
- Attributes – Data fields containing information within requirement elements.

## Getting Started with Requirements Management

Gathering requirements is typically the first step in developing a solution, be it for the development of a software application or the detailing of a business process. Requirements are essentially “what the system needs to do”. The requirements management built into EA can be used to define requirement elements, link requirements to model elements which implement them, structure requirements into a hierarchy and report on requirements.

Before we get started, you will need to open a project within which you can work. Most of the examples given below are related to the [EAExample.eap](#) model provided as part of the Enterprise Architect installation. The example model is installed under the Enterprise Architect Program Files directory:

..\Program Files\Sparx Systems\EA

## Requirements Modeling

In Enterprise Architect you can capture or view requirements directly in the model, using the Requirement element available from the Requirements palette of the Toolbox. This is known as an external requirement.

- **The Requirements palette is located at the far left of the Enterprise Architect interface, in the Toolbox.**

The use of a UML element allows the requirements to be traced to other UML elements such as other requirements, use cases, test cases, and analysis or design elements. This element can be used to model or document any requirements ranging from formal business requirements through to performance or security requirements. These requirements are shown diagrammatically with their relationships (see Figure 1). The core information behind any one requirement is defined in the properties section (see Figure 2), user defined “attributes” can be created using Tagged Values and Profiles (see [Predefining Tagged Values Types for Requirements](#))

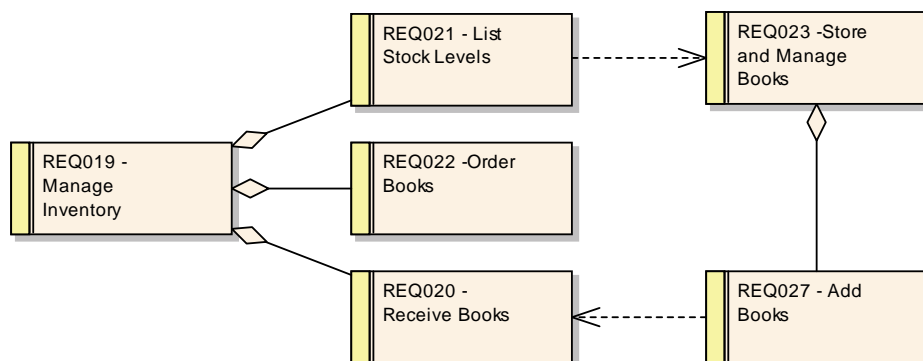


Figure 1: Custom diagram showing the requirement element

### Creating an External Requirement

1. Left-click the **Custom Button** in the **UML Toolbox** to open the custom palette
2. Click-and-drag the **Requirement** element from the palette onto the diagram
3. Enterprise Architect allows you to specify several properties of the requirement
  - a. The **Short Description** field will be displayed on the diagram
  - b. See **External Requirement Attributes** below for more information
4. Click the **OK** button once finished
5. These properties can be edited again after creation by double-clicking the requirement element
6. The new requirement should now be visible in the **Project View**.

### Common Uses

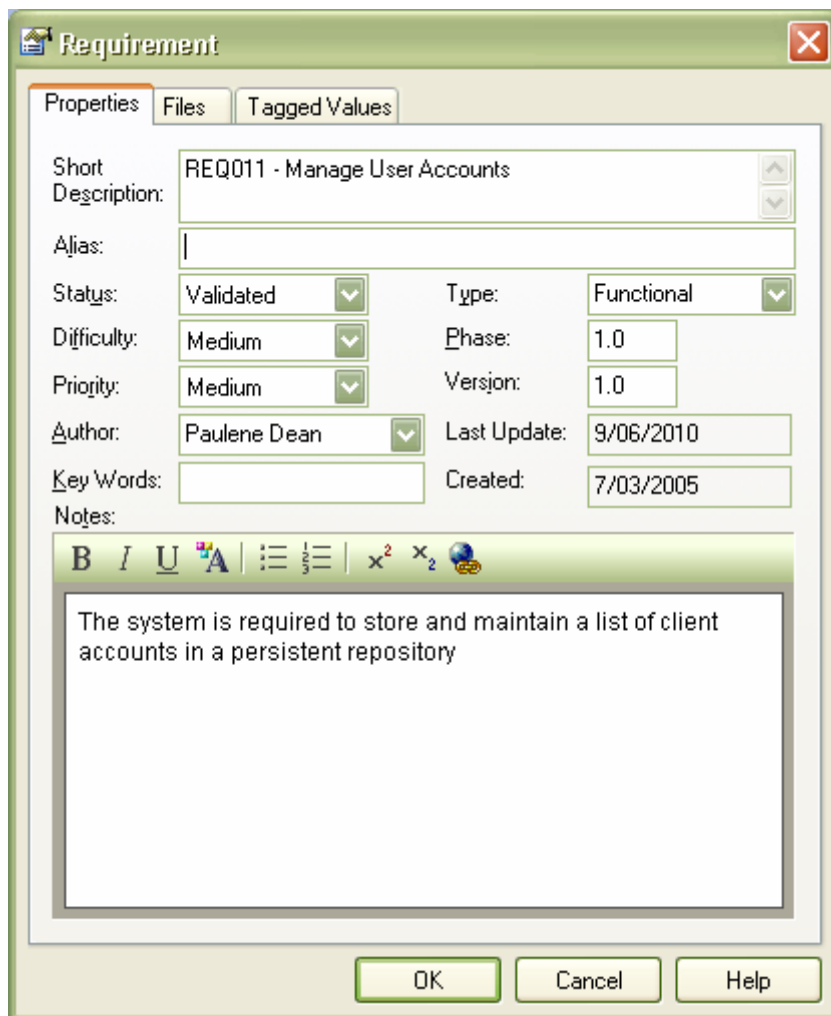
- Modeling business requirements
- Modeling stakeholders requests

### Tips and Tricks

- The requirement elements name can be kept simply as text, or it can be manually numbered along with the text label. Enterprise Architect supports auto-numbering of requirements (see *Auto-Naming Elements* below).

## External Requirement Attributes

Every element, including a requirement element, that is part of a model has properties or attributes. In Enterprise Architect these are assigned in the properties sheet. (*Double-click on the Requirement*). Enterprise Architect has built-in requirements attributes such as status, difficulty, priority, and type. An example of the properties for a requirement is shown below.



**Figure 2: External requirements properties**

### Changing the Properties of an External Requirement

1. Double-click on the requirement element (either on the diagram or in the project view).
2. Alter the properties displayed in the properties window.
3. Click **OK** to apply the changes and exit the properties window.

### Useful Features for External Requirements

When using external requirements, there are a number of ways of customizing the data and viewing the details. The following are some features, when used in conjunction with the external requirements, which enhance their use.

- Creating user-definable attributes using tagged values
- Viewing Requirements using the Element List view or diagram view
- Setting up relationships between requirements, as well as relationships to other UML elements such as use case, classes, test cases, etc.
- Tracing relationships between requirements and other elements.

- Creating a requirement hierarchy using child elements or packages

The following sections cover the details of these options.

### Adding Additional Attributes to Requirements

It is common that there are a series of requirement attributes specific to any project. You can enter any number of additional attributes such as stability, cost, and lateness penalty through the use of tagged values.

Tagged values can be defined on a one-off basis for any element, or predefined to be included on creation of a new element.

Tagged value data for an element is available as a separate window, which is accessed using **Ctrl+Shift+6** (or from the main menu **View | Tagged Values**).

See Figure 3 for a diagram showing a one-off addition of a tagged value.

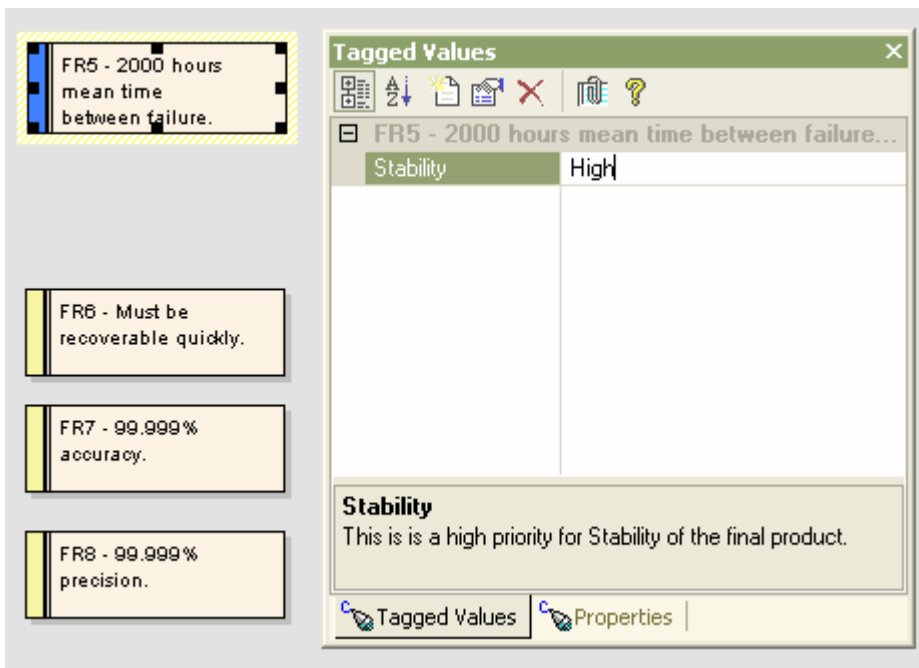


Figure 3: Requirements tagged value sheet allowing the assignment of attributes

If you use tagged values often, consider leaving the window open and docked (see Figure 7).

### Predefining Tagged Values Types for Requirements

Elements in Enterprise Architect, including requirement elements, can have an extended set of attributes defined for a project. The element attributes can be predefined using either a *UML Profile* or an *EA Template*. See Figure 4 for an example of an element using a predefined set of tagged values for a project's requirement elements.



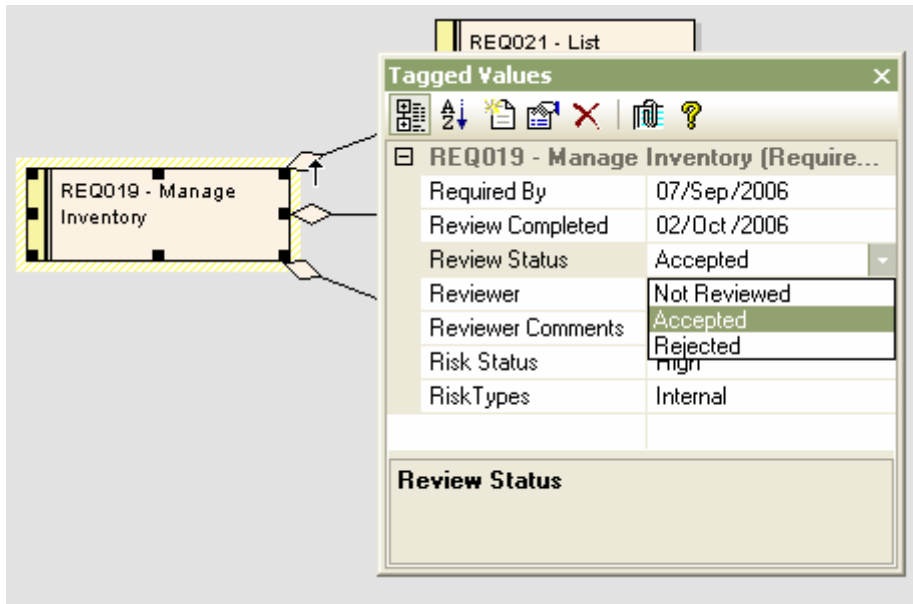


Figure 4: Using predefined tagged values

The predefined tagged values types can include a number of standard formats, such as date/time, calendar view, drop-down lists, etc.

These extended attributes can also be viewed directly on the element in the diagram.

To set this mode for a specific diagram, right-click on the diagram, and in the context menu, select: **Properties | Elements | Show Compartments | [✓] Tags**. Below is the same element in Figure 4 viewed in this mode.

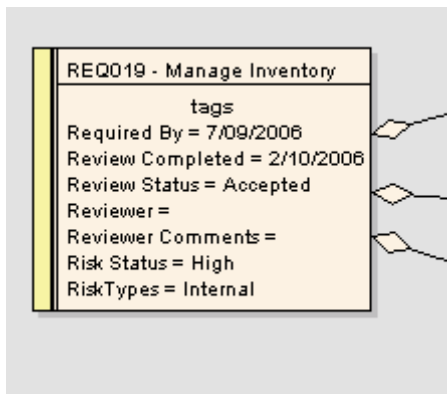
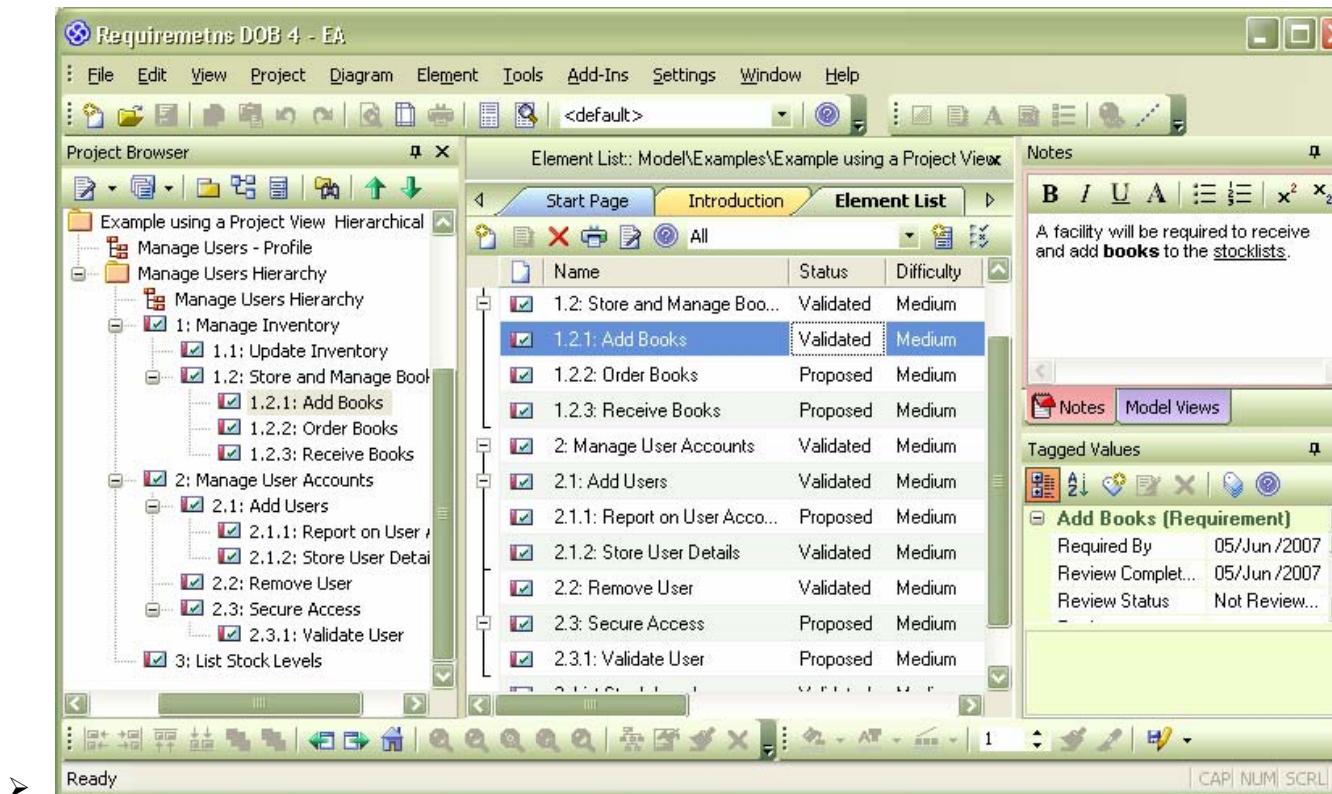


Figure 5: Tagged values visible on elements

For more information on extending requirement attributes using tagged values see:

- Appendix [Defining Requirement Attributes using a Profile or a Template](#)



- Figure 7 for a view of the docked tagged values window used with the Element List view.

### Element Numbering

Along with the Package hierarchy, Enterprise Architect also supports creating a hierarchy of elements under a package. Element Numbering combined with this hierarchical structure, allows for elements within a Package to be numbered in a 1.1.1 format. This feature can be set on any package and applies to the Elements contained in the root of that Package (it does not apply to child packages).

Below is an example an Element hierarchy with the Element Numbering set on:

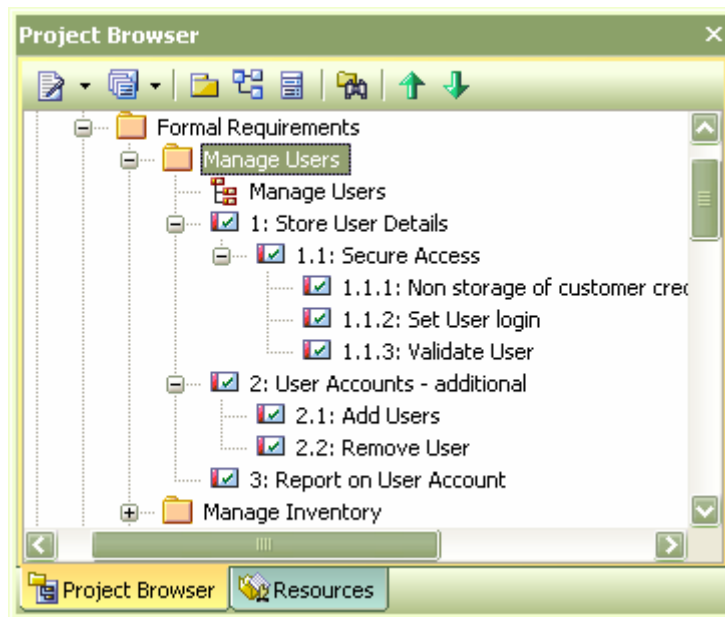
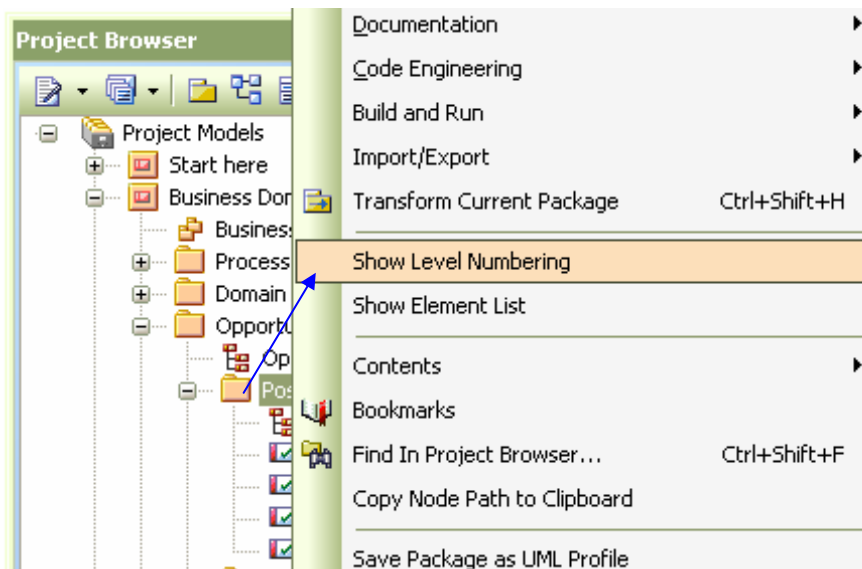


Figure 6 Shows an Element-Tree of Requirements with the Element numbering set on.

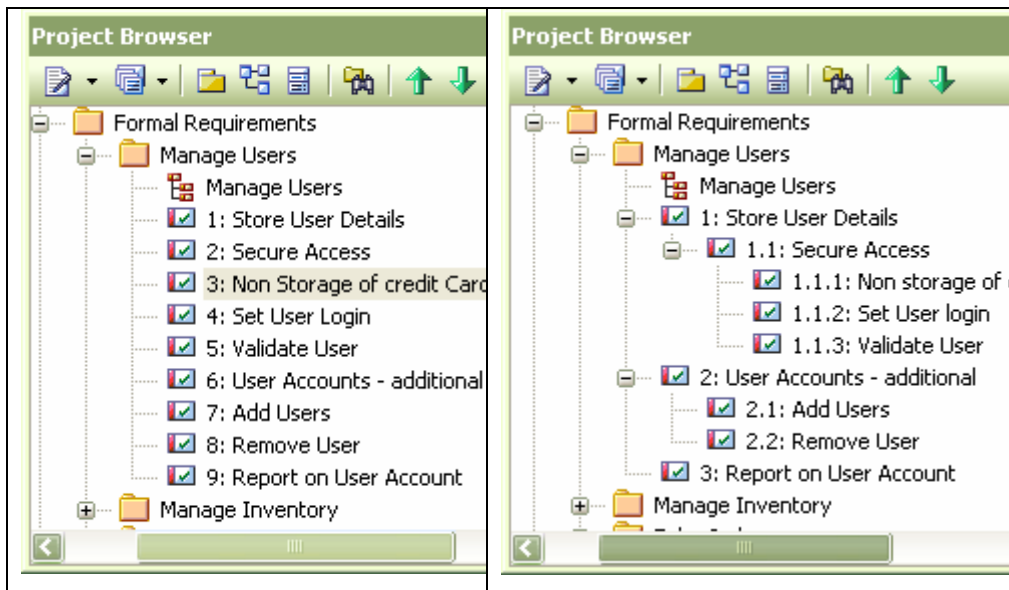
**Figure 6: An Element Hierarchy with Level Numbering**

To enable this option:

- Select a **Package** in the **Project Browser**
- **Right-click** and from the context menu select: **Show Level Numbering**:



From the **Project View**, Elements that are numbered can be re-ordered simply by dragging a child element onto another element. Child elements are re-numbered to fit the parent elements 1.1 format numbering. Below is an example:



**Note:** This numbering can be reported in the RTF report generator using the Element Section – “LevelNumber” field: {Element.LevelNumber}

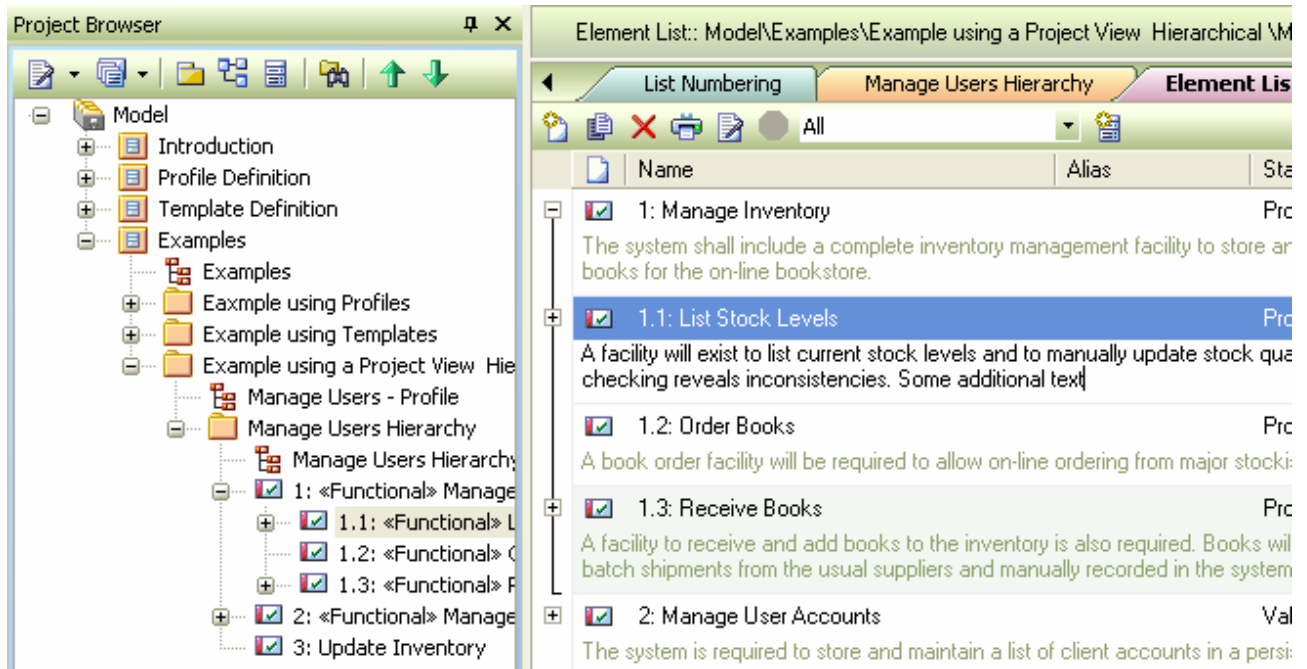
### Different Views of Requirements Using the Element List View

It is common that requirements are defined by users that are un-familiar with UML diagrams. Enterprise Architect supports a text-based view of the requirements, while maintaining a hierarchical structure in the Project View.

Some features to use are:

- The **Project View**. This can be docked on the left of the screen to display the hierarchy.
- The **Element List View**. The diagram view can be set to text view mode.
  - To swap between the diagram view and the Element List view from the main menu, select **View | Element List**.
- The **Notes** and **Tagged Value** windows can be set to be viewed by default.
  - To view the notes window select **View | Notes**.
  - To view the tagged values window select **View | Tagged Values**.

The following is an example of viewing the requirements in this text-based mode.



Below is a text-based example with the Tagged Value window and the Notes windows open:

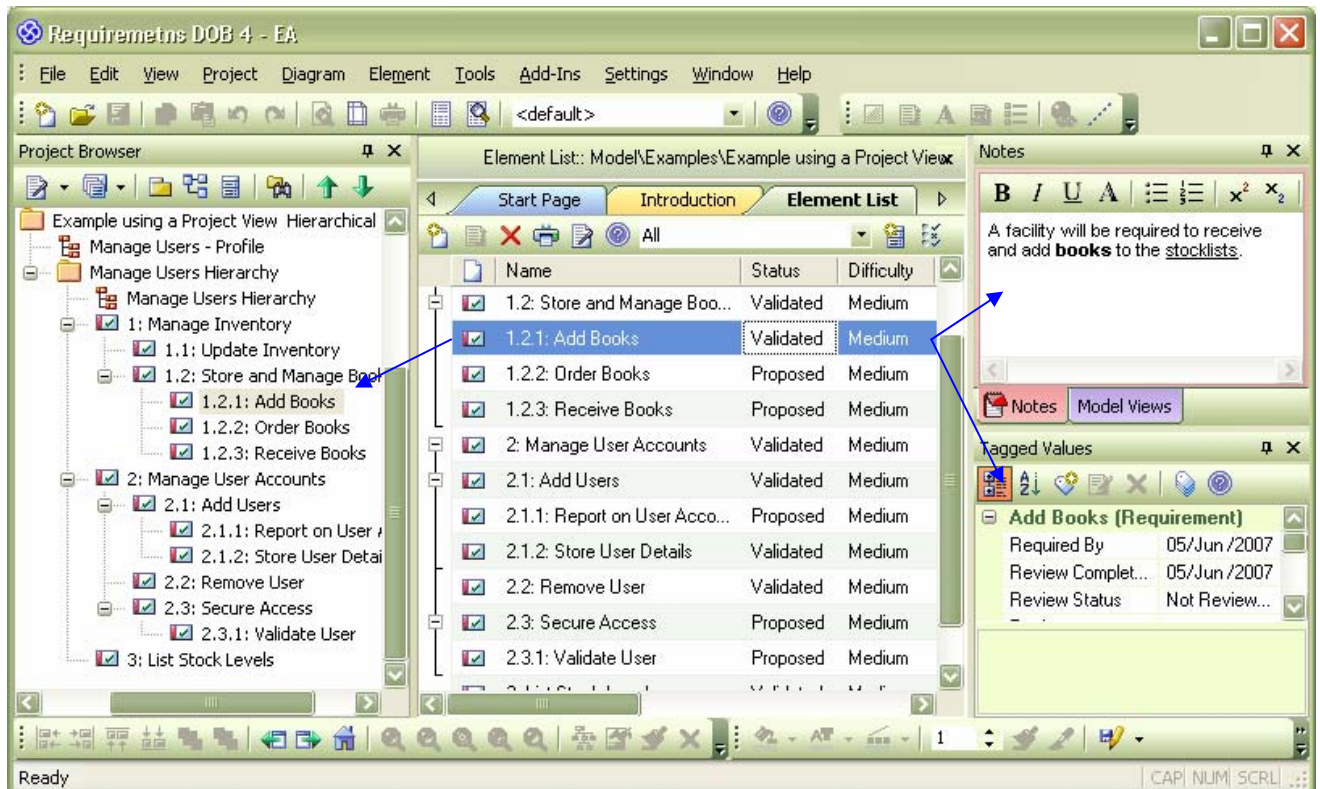




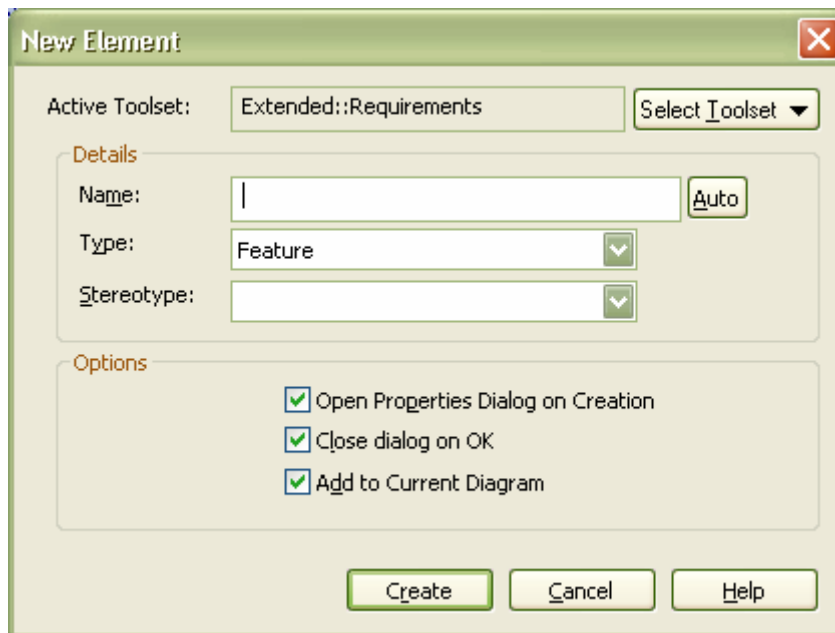
Figure 7: An alternate visual layout

## Tips and Tricks

- The **Project View** can be set to hide the Stereotype (eg. <<Functional>>) using: **Tools | Options General | [ ] Show Stereotype**. The view above has this option set off.
- Using the  options icon opens a dialogue for setting:
  - Nested Packages to be viewed (recommended).
  - **Element Notes** to be viewed under each element in the Element List.

### Adding New Elements

When using the Element-List view, new Elements are added via the Insert Element icon . This opens the following dialog:



The 'New Element' dialog box is shown with the following fields and options:

- Active Toolset:** Extended::Requirements (with a 'Select Toolset' dropdown arrow)
- Details:**
  - Name:** (text input field with an 'Auto' button)
  - Type:** Feature (dropdown menu)
  - Stereotype:** (dropdown menu)
- Options:**
  - Open Properties Dialog on Creation
  - Close dialog on OK
  - Add to Current Diagram

Buttons at the bottom: Create, Cancel, Help.

To create the new element; just add text to the **Name** field and Press **OK**. By default, this will open the Element Properties dialog for entry of detailed text such as notes etc.

## ***Traceability and Relating External Requirements***

In EA, there are three key methods used for tracking and forming relationships between external requirements and other elements. These relationships will define how those requirements are to be implemented within the system. The three key methods are as follows:

- **Creating and viewing relationships using diagrams**  
Relationships between elements are easily created in a diagram using standard relationships defined in the toolbar.
- **Creating and viewing relationships using the Relationship Matrix**  
The relationship matrix provides a process for viewing or creating links between elements in different packages, independent of them being defined in a diagram.
- **Tracing relationships using the Traceability View**  
The hierarchy window provides a feature for tracing all the relationships of a selected element.

### **Creating Relationships Using Diagrams**

Creating relationships between elements on a diagram is a simple process in Enterprise Architect. A click-and-drag method from source to target is used to define these relationships.

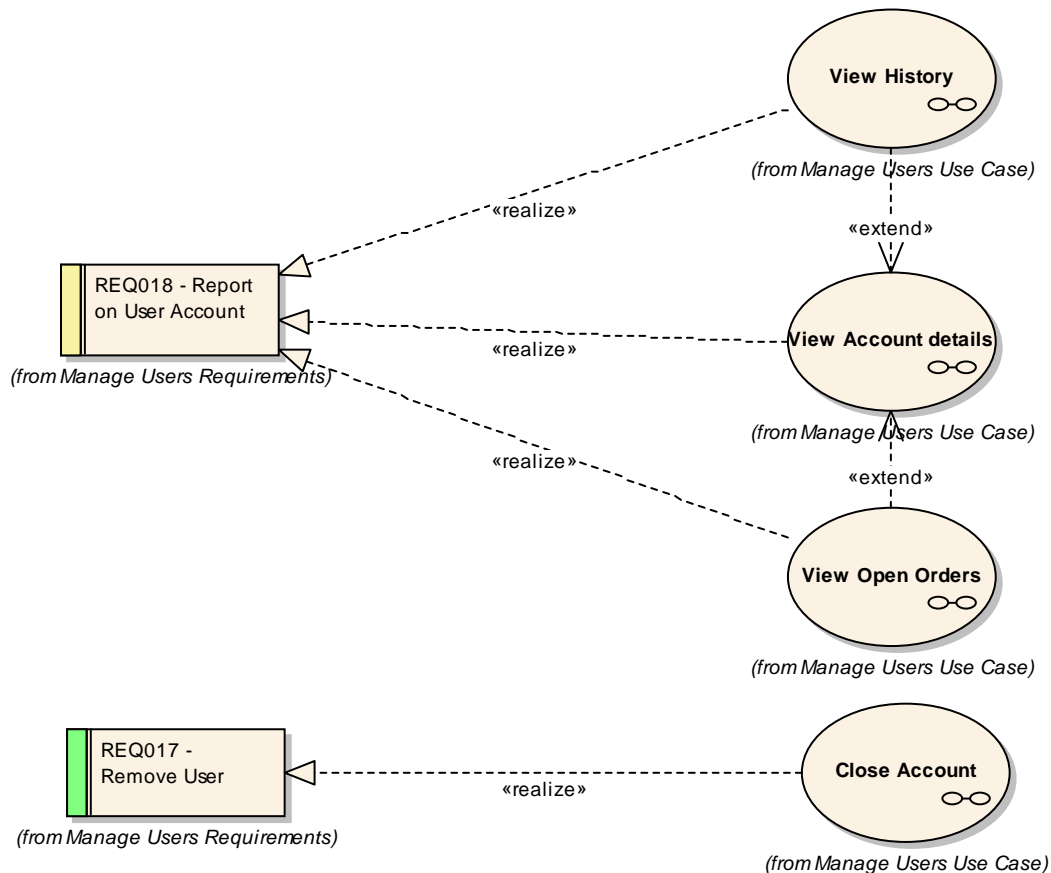
1. Add two requirements elements to your diagram.
2. Select the type of relationship you wish to create from the lower half of the custom palette (eg. **Aggregate**).
3. Click-and-hold the left mouse button on the source element for the relationship.
4. Drag the mouse to the target element, and then release the button.

### **Creating a Common Diagram**

Creating links between objects in different packages can be a simple process using a common diagram. To do this, simply:

- Create a new diagram
- Drag onto the diagram, from the **Project View**, the elements in the different packages.

Below is an example of a diagram with elements from different packages that were linked via the relationship matrix.



**Note:** The properties of this diagram have been set to display the diagram source (using the **Diagram Properties: [✓] Highlight Foreign Objects**).

## Types of Relationships

### Aggregation

Requirements linked by aggregation relationships form a composition hierarchy. High level requirements may be composed of lower level requirements, which in turn are made up of finer and more specialized requirements. This hierarchical structure helps manage the complexity of large systems with thousands of requirements and many elements being employed to implement the requirements.

### Realization

Requirements are implemented by model elements, such as use cases, classes, interfaces, components, etc. You may specify this relationship in EA using the realization link. A model element is marked as 'Realizing' a requirement. Once this link exists, EA will display the requirement in the element responsibilities tab, in the requirement Traceability view, and in the dependency and implementation reports, as well as the standard RTF output (See [Requirements Documentation \(Reports\)](#) below for more information on reports).

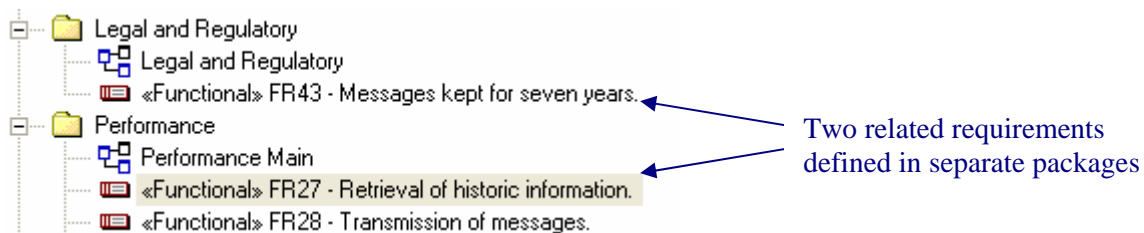


## The Relationship Matrix

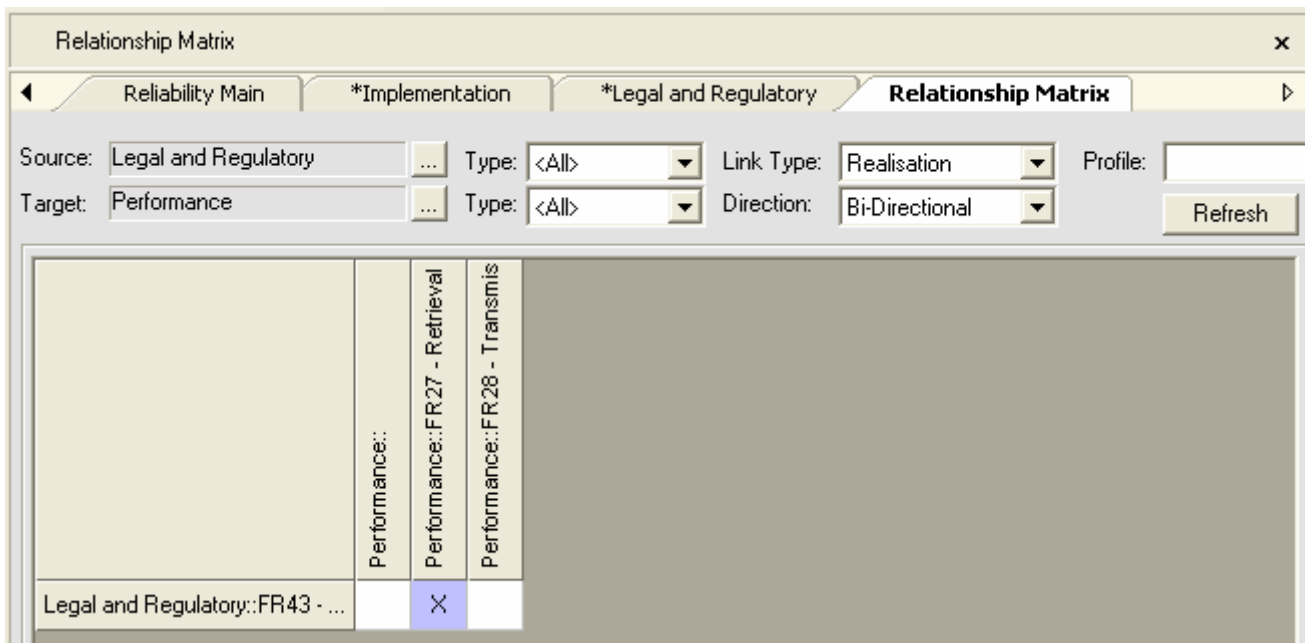
The relationship matrix allows you to create and view relationships, regardless of what diagram or packages the elements are placed in. It can be used with any UML element, but it is particularly useful with requirements management for two reasons:

- 1) With a large system definition, there will be requirements defined in different packages and diagrams that may have interdependent relationships. The relationship matrix can be used to set up these relationships that would otherwise need to be defined manually in a diagram.
- 2) As the development phase progresses, each element under the requirements, such as use case, classes, etc., needs to have its root defined by a specific requirement or group of requirements. It is this linking that is critical to backward traceability.

Below are two related requirements that are in separate packages.



To view the relationship matrix, change the view (**View | Relationship Matrix...**) from the menu at the top of the screen. The source and targets are set up to show the 'Legal and Regulatory' package as the source and the 'Performance' package as the target.



### Adding a Relationship between Requirements Using the Relationship Matrix

1. Navigate to, and open (double left-click), the functional requirements diagram.
2. Add a requirement with the description "Font color must be green".
3. Navigate to, and open (double left-click), the non-functional requirements diagram.
4. Add a requirement with the description "Must match current look and feel". Because these two requirements are related, the relationship should be defined.

5. Select the relationship you wish to create from the **Link Type** options (eg. Aggregate).
6. Right-click on the area where the two requirements overlap and select **Create new relationship...**
7. Right-click again for further relationship options.

### Common Uses

- To view the relationships that exists between packages and elements.
- To create the relationships between requirements not located together in a diagram.
- To view the impact of a requirement change request on the system.

### Tips and Tricks

- Use the relationships window to create, edit and delete relationships, rather than doing this graphically in the model diagrams.
- Save your favorite or commonly used matrix profiles. These will then appear in the resources view. This is very useful because it may often be necessary to look at the same kind of relationships.
- EA also supports an automatic process of creating a relationship using the drag-and-drop – see [Drag and Drop Realizations](#) in the **Additional Requirements Management Features** section.

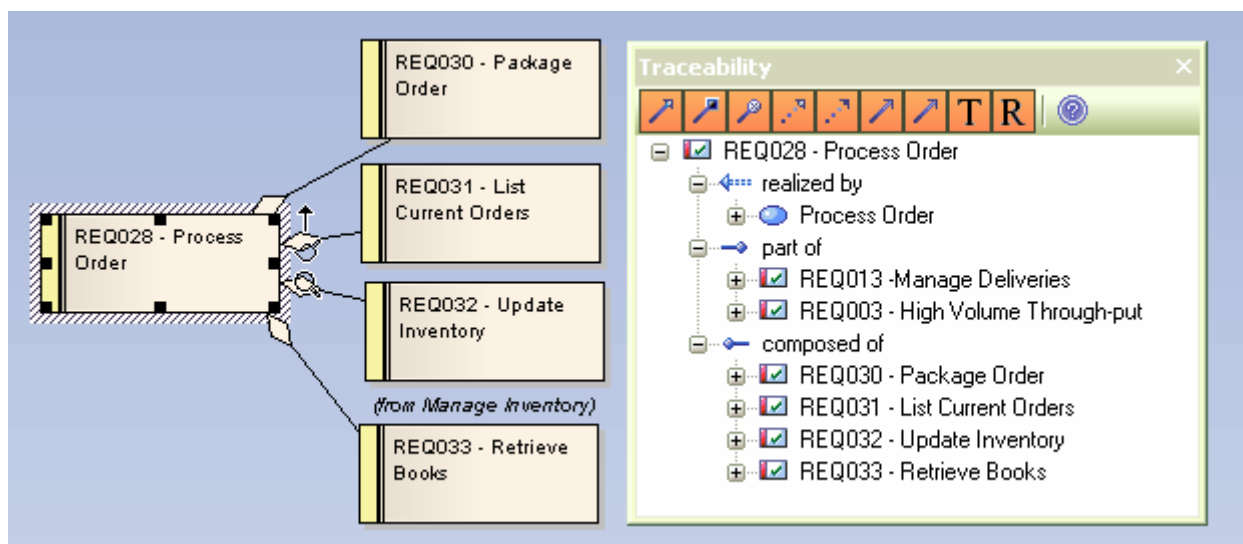
## Using the Traceability Window

The Traceability window allows you to view the relationship between requirements. It can be used with other UML elements, and is particularly useful in requirements management.

### To Use the Traceability Window for Viewing Relationships

- Open the Traceability window (**View | Traceability** or **Ctrl+Shift+4**).
- Select an element to display its relationships.

The elements relationships are then displayed in the Traceability window.



**Figure 8: The Traceability window showing the way requirement relationships are listed graphically**

The Traceability window is particularly useful to see the relationships from requirements to use cases, and down through the different levels of UML diagrams. Below is an example of relationships between the requirements, and the use case for 'Processing an Order'.

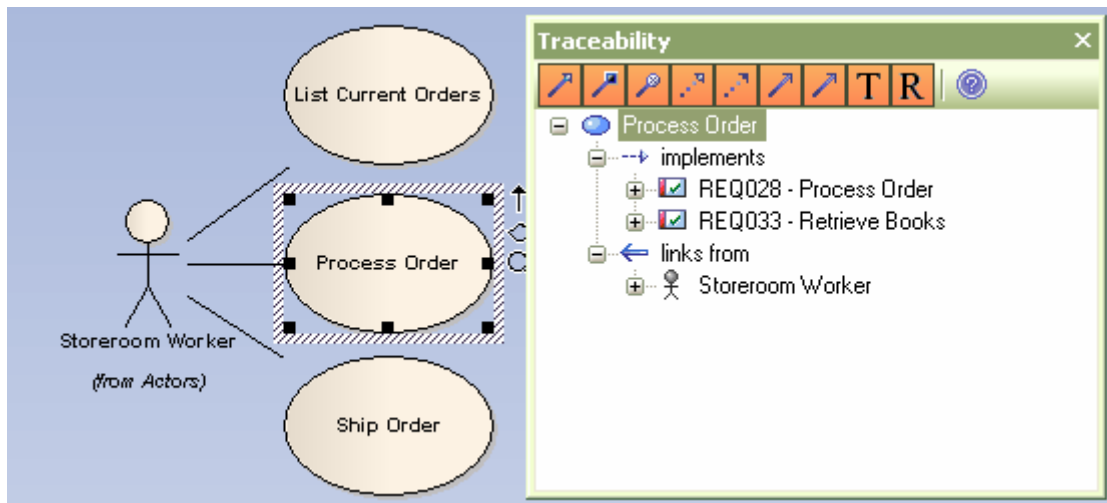


Figure 9: The Traceability window showing the way use case relationships are related to requirements

### Common Uses

- Often a diagram is deliberately drawn to show only one aspect, or part of, the underlying model. The Traceability window is particularly useful to show the related elements that are not visible on the diagram.
- To get a quick snapshot of how a requirement (or any other element) relates to other elements in the model.

### Tips and Tricks

- While the Traceability window gives a view of the relationships for a single element you may consider using the relationship matrix as a way of viewing relationships at a higher level.

## Change Control

Enterprise Architect supports features for monitoring changes to requirement definitions. These include Auditing, managing Baselines, Element Change requests and Issue logging.

### Auditing

The Audit feature enables you to record model changes in Enterprise Architect. It records details of **who** changed an element, **when** and **what** was changed, along with the prior state of the model. This can be particularly useful for recording a history of changes to requirements models.

To enable this feature:

1. From the main menu select: **View | Other Project Tools | Audit View**, which opens the following view:



2. Select the **Audit Settings** button.
3. This will bring up the Audit Settings window:

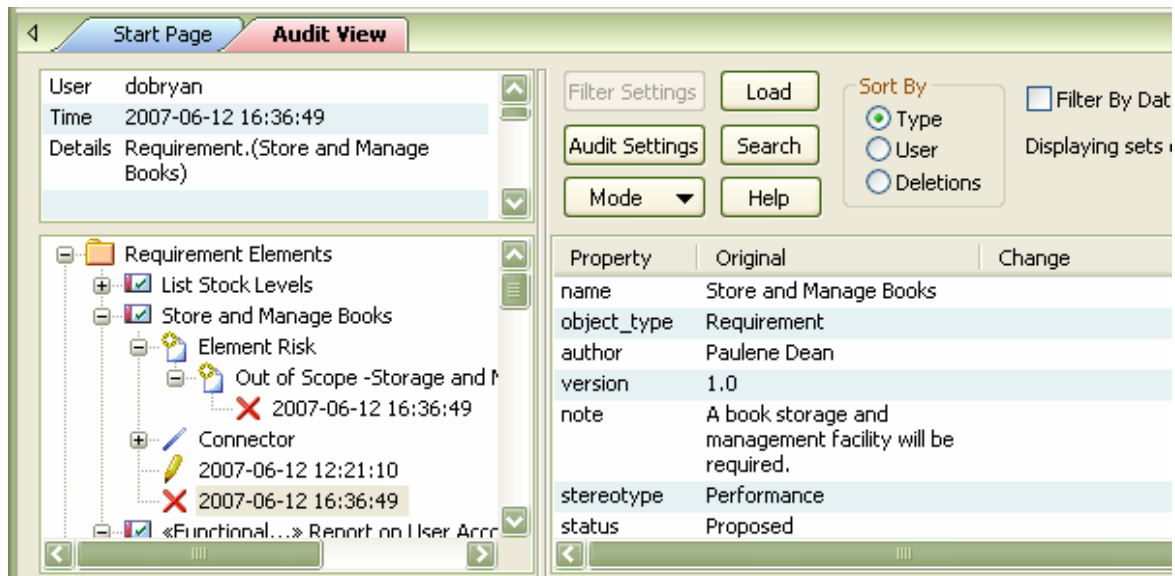


4. In the Audit Settings window set:  **Enable Auditing** on as shown above.

The example below has the **Element List** view option set on. The **Output | Audit History** window shows the list of changes for the selected Element:

Timestamp	Columns	Before Change	After Change
2007-06-08 17:04:10	name	REQ011 - Manage User Accounts	REQ011 - Changed Manage User Accounts
2007-06-07 15:52:07	note	The system is required to store and maintain ...	The system is required to store and maintain a list of ...

The full history of changes is viewable by Element-type using the **Audit View**:



For more information on using the Auditing feature see Enterprise Architect's help under: [Projects and Teams](#) | [Change Management](#) | [Tracking Changes](#) | [Auditing](#)

## Using Baselines

The auditing feature outlined above, provides continuous tracking and logging of changes to requirements. The [Baseline Management](#) feature provides additional support for comparing and merging changes. It allows Baselines of a model to be created on a periodic basis (i.e. by month, phase, version or build). Baselines can then be compared to the current model state and changes selectively merged.

For more information on setting up baselines and viewing differences see the help section: [Projects and Teams](#) | [Change Management](#) | [Tracking Changes](#) | [Package Baselines](#)

## Change Requests and Issues on External Requirements

Enterprise Architect supports logging of Change-requests against Requirements. This can be defined using two different methods:

- Using the [Maintenance View](#) to list Changes, Defects, Issues and Tasks against each element.
- Using custom elements of type "Issue" and "Change" linked to the External Requirements being altered.

Each has their different uses which are outlined as follows:

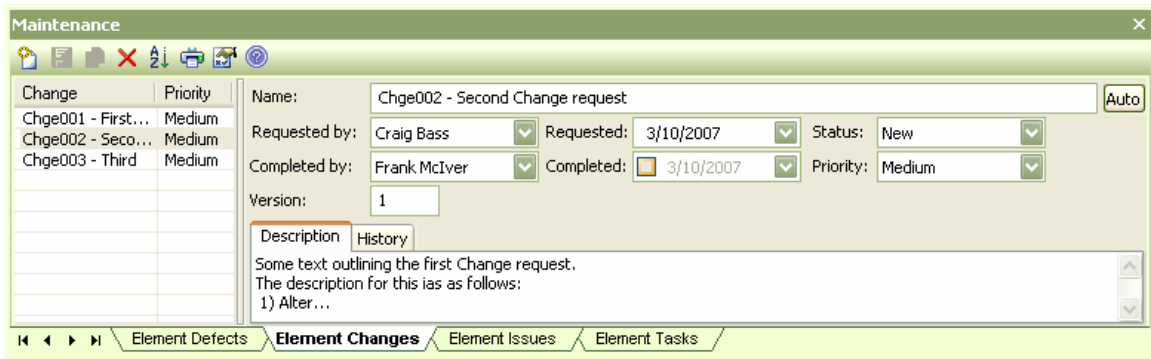
### Using the Maintenance View

The [Maintenance View](#) can be used to log changes against any Element or Package. This provides listings for:

- Element Defects
- Element Changes
- Element Issues
- Element Tasks

These include fields for recording "by whom and when" the request was made and completed, as well as Status, Priority, Description and History.

The **Maintenance View** can be accessed from the main menu using: **View | Other Element Tools | Maintenance** or **(Alt+4)**. Figure 10 is an example of a set of changes listed for an element:



**Figure 10: Maintenance View**

The common use for requirement management is for logging detailed Requirement-Issues and Change-Requests against any existing Requirement Element.

### Using Maintenance Elements for Changes and Issues

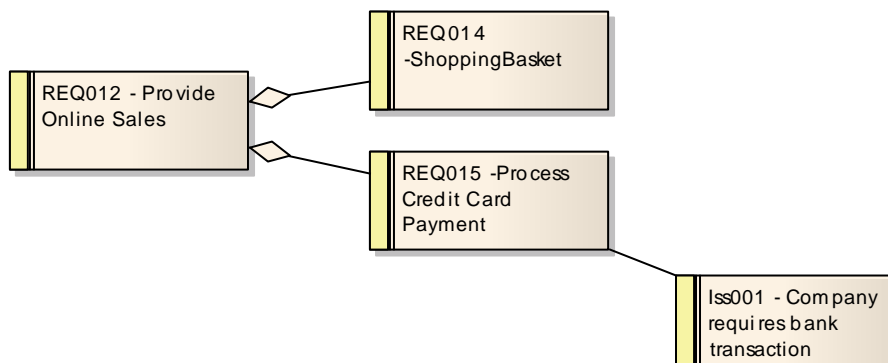
Enterprise Architect’s maintenance elements include Elements of type: **Issue** and **Change**. These are accessible from: **Toolbox | Maintenance** or **Toolbox | Custom**.

Maintenance elements can be linked using a connector to any element (i.e. an External Requirement) to display a change or an issue.

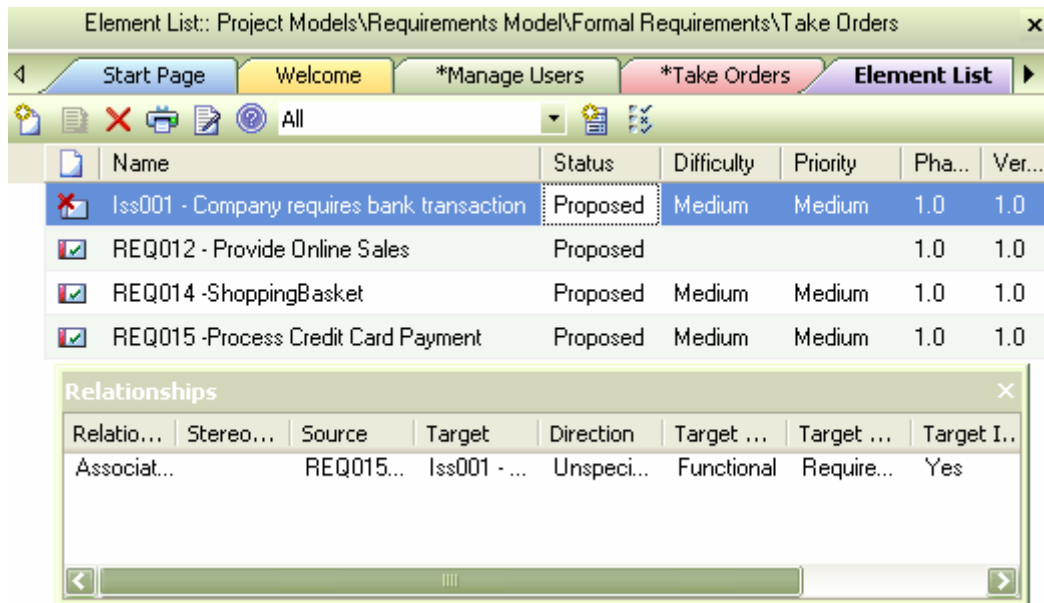
#### Tips:

- These Elements can be stored in the package containing the associated requirements or in a separate package containing a set of changes.
- They can be linked to Requirement Elements in common diagrams or using the Relationship Matrix.
- These Elements can be customized as part of a Profile to include extended properties.

The following diagram illustrates the use of an Issue Element associated with a Requirement.



Below is the same data viewed in the **Element List** view along with the **Relationship** window showing Elements related to the selected Issue:



## Internal Requirements

Enterprise Architect allows you to enter requirements at the level of an individual UML element. At this level these requirements can best be thought of as the 'responsibility' of the element.

Multiple internal requirements may be defined within any element from the properties window (**double-click the element**). The following diagram displays a single requirement defined within a use case element.

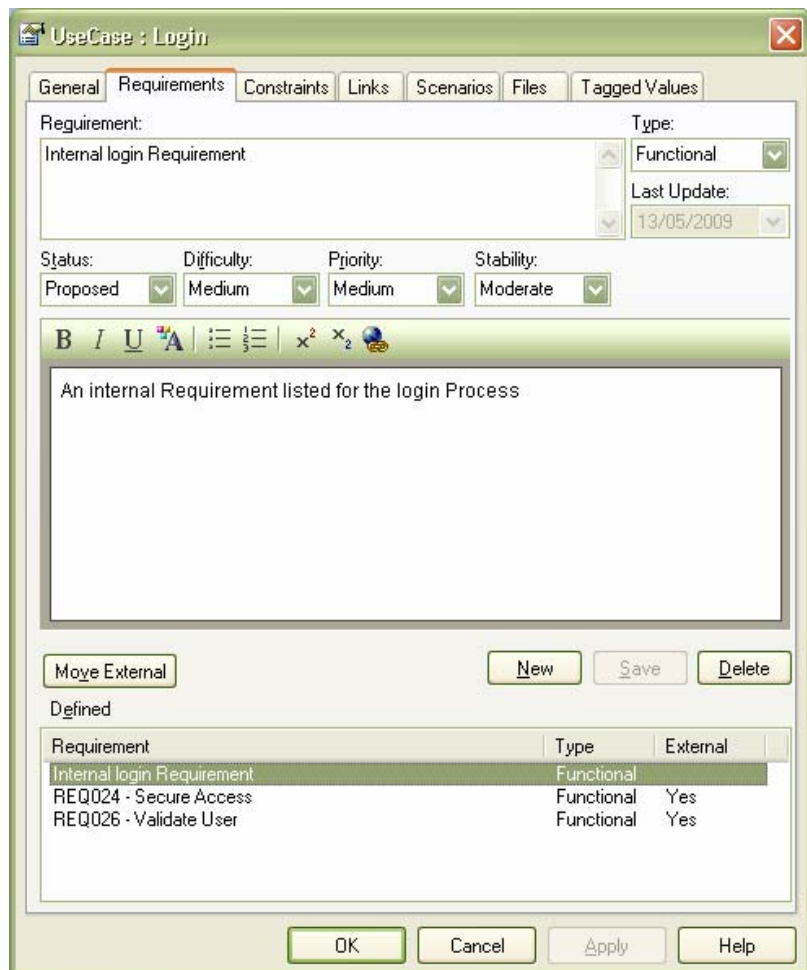


Figure 11: The property sheet for a UML element showing how to create internal requirements

**Internal requirements can be externalized** (see the [Move External] button above). This will create a new external requirement element with a realization relationship back to the original element (in this scenario – a use case element).

The definition of internal requirements within UML elements, such as use cases, gives a simple introduction into the more complex requirements definitions, using UML modeling in general.

### Common Uses

- This can be the scenario used for setting requirements in general within EA. There are numerous reasons for defining requirements using the external requirement elements.
- To view the all the requirements or responsibilities of the element.

### Tips and Tricks

- Even if the element doesn't have internal responsibilities, it will typically have external requirements. These will be displayed in the list, with the column 'External' displaying 'Yes'. Anyone working with the element will be able to view these and get an immediate view of what the element is required to do.
- As stated above, while working with an element you may define an important internal requirement. To ensure that this is captured and included, you can optionally move the requirement external (by using the move external button). This creates a custom requirement type and will request a package that the new requirement should be placed in.



## Creating Customer Quality Requirements Documentation

The definition of requirements is often used as a contract either between different departments within an organization or between organizations. Therefore, it is often required that high quality documentation of this can be generated.

### *External Requirements Reports*

Enterprise Architect's RTF reports generator includes a report for external requirements. This can be easily copied and modified, using the report generator, to suit your reporting needs.

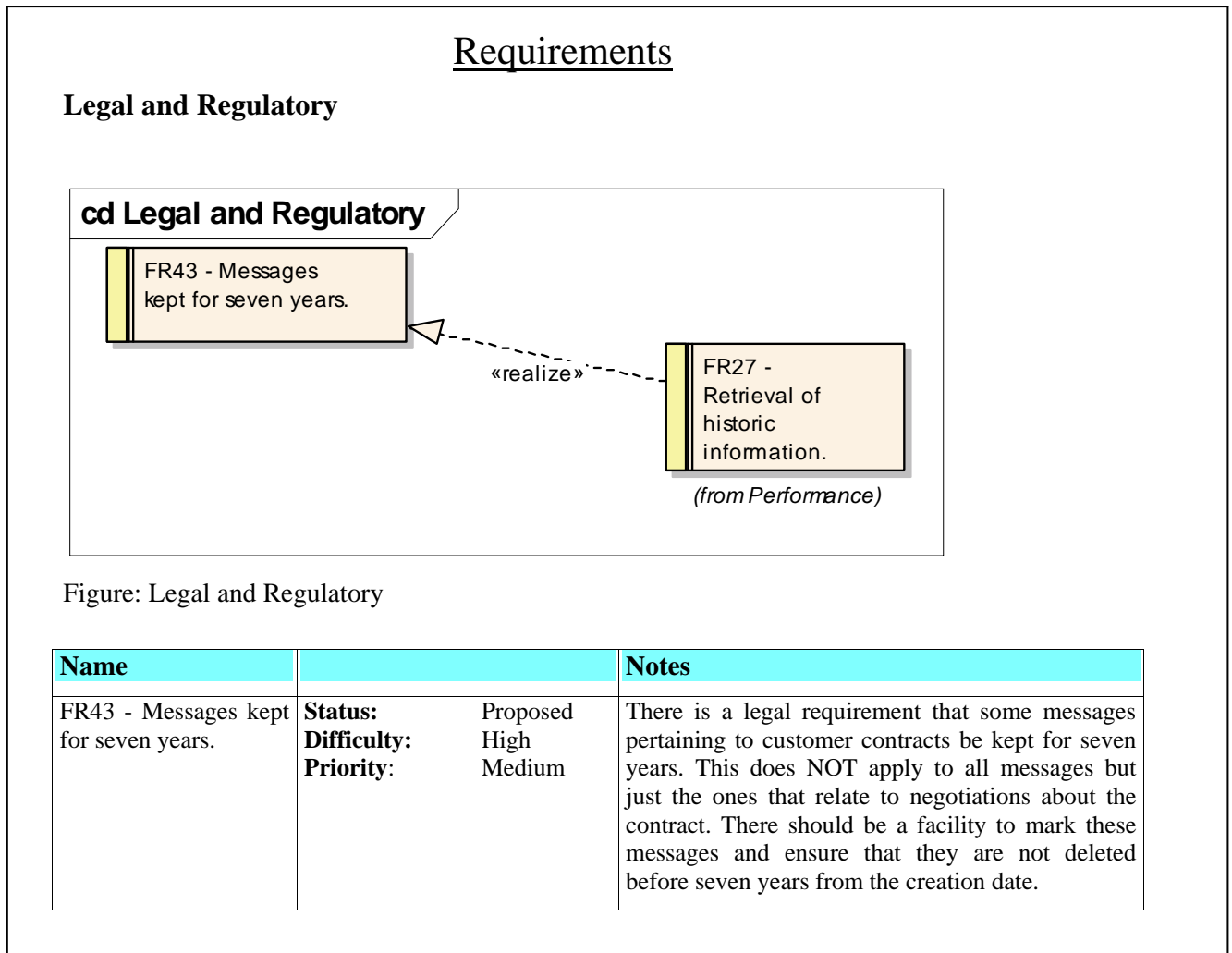


Figure 12: A short example of the RTF requirements report

### To Copy an Existing RTF Document Template for Editing

1. From the main menu, select **Project | Documentation | Rich Text Format Report** (or press **F8**).
2. In the **Generate RTF Documentation** window, select **Manage Templates**.
3. In the **RTF Templates** window, select **New**.
4. In **New RTF Document Template** window, select from the **Copy Template** drop-down {Basic Template}.
5. In **New Template** type in a new name for your report.

This will open the editor ready for you to start editing the existing template to create a new report template in your own format.

## Internal Requirements Reports

If you want a report on the internal requirements, the {basic template} report includes a section for the internal requirements. It is simple to copy this and remove the major detail around the internal requirements to give a report focused on these.

To modify this follow the procedure outlined above.

### Common Uses

- To provide high quality customer documentation.

### Tips and Tricks

- Consider making structural changes to these documents using the RTF report editor.
- Consider generating in your report definition a glossary, as an inclusion with the formal documentation.
- Consider encouraging customers to see the EA model as the documentation. (With the free EA viewer, you can allow customers to view the EA models.)

## *Implementation Report, Dependency Report, and the Element List View*

There are two additional reports, as well as the Element List view, that are very useful when managing requirements. They can be used in any discipline, but are particularly useful in the requirements discipline.

### Implementation Report

The implementation report shows:

- Lists of elements that can be realized by other elements in the model; and
- Other model elements that realize them.

To access the implementation report, select from the main menu [Project](#) | [Documentation](#) | [Implementation Details](#).

### Common Uses

- To locate all elements that should have realizations.
- To locate all elements which implement a particular element.

### Tips and Tricks

- Enterprise Architect, by default, only lists commonly realized elements such as use cases, requirements and components. By choosing [Set Target Types](#) you can tell EA to report almost any element.

### Dependency Report

The dependency report lists the elements that have a dependency on another element. This is very useful for checking the dependencies placed on requirements. To access this, select from the main menu [Project](#) | [Documentation](#) | [Dependency Details](#).

### Element List View

The Element List view can be used to get a quick and simple, but clear, picture of the requirements and their detailed text. The Element List view swaps the diagram between two modes.

- The diagram mode which displays the elements in a graphical format.
- The Element List view mode which displays the textual content of the elements.

To swap between these two modes, select from the main menu: [View](#) | [Element List](#).

This can be used simply, as displayed below.

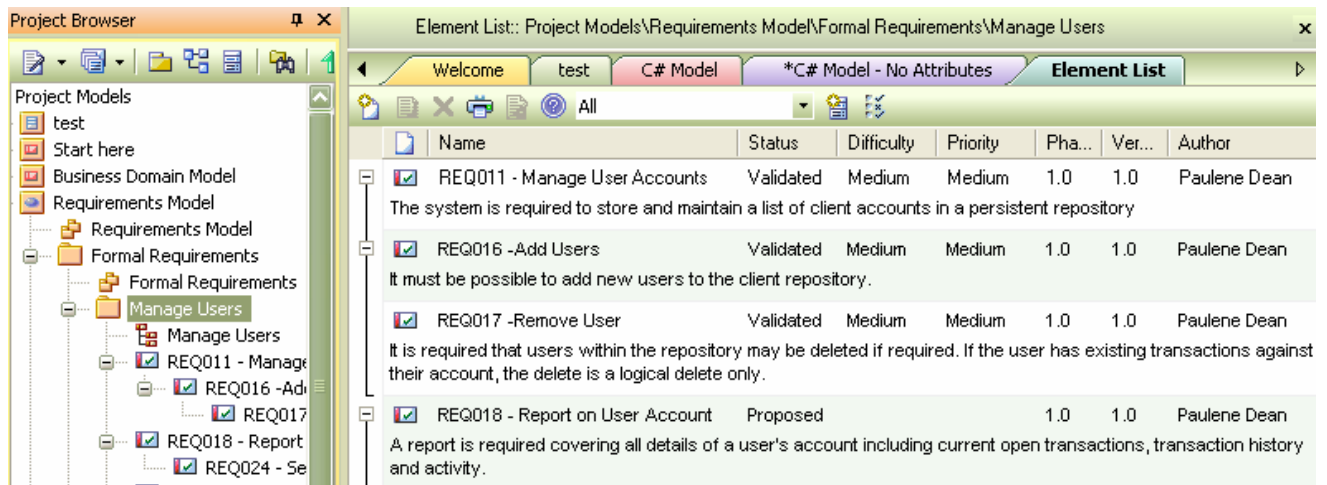


Figure 13: Screen view in report mode

There is also the option to create RTF reports or directly print text reports from this.

RTF reports:

- Select a set of elements to report (or Ctrl-A to select all)
- Right-click on the selection.
- Select: **RTF report | Generate one Report for all Selected.**

Print reports:

- Right-click on the listing and from the context menu select: **Print.**

## Additional Requirements Management Features

Enterprise Architect provides a number of other features for requirements management that will be covered in the following section.

### Creating Your Own Requirement Types

Enterprise Architect provides you with a number of default requirement types. You are able to modify these, add your own, or even completely tailor the list to your own project or organization's needs. (This is accessible from the main menu: **Configuration | General Types | Requirements**). It is displayed as follows:

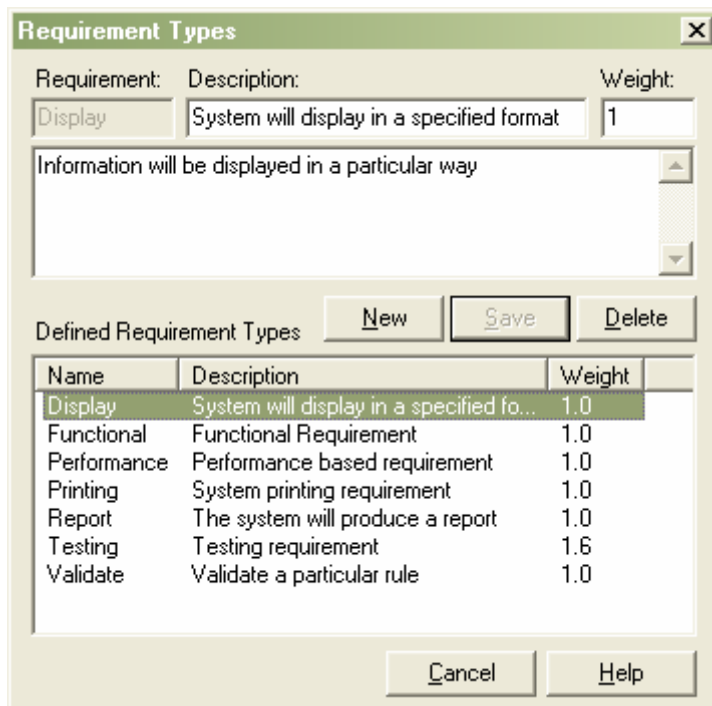


Figure 14: The requirements type definition window showing how to define your own requirement types

### Common Uses

- Use this for complying with an industry, organization, project process, or standard that prescribes a list of requirement types. For example, the IEEE's Guide to Software Requirements Specifications.

### Tips and Tricks

- Consider re-using the requirement types from a previous or related project.

## Color Coding Requirements

External requirements may be color coded to enable quick visual cues indicating the status of a requirement. To enable color coded external requirements take the following steps:

1. From the menu, select **Tools | Options** to open the local options window.
2. From the hierarchical tree select **Objects**.
3. Check the **Show Status colors on diagrams** checkbox.

### Common Uses

- Gives a clear diagrammatic view of the status that each requirement has reached.

## Auto Element Naming

This feature allows you to configure Enterprise Architect to automatically name and number requirements as they are created. (It can be used with almost any UML element such as use case and classes). It is particularly useful with requirements, since these are often required to have a unique reference for external checking.

To enable auto element naming, take the following steps:

1. Select **Settings | Auto Counters...** from the menu to open the **Auto Element Naming** window.
2. Change the **Type** to requirement.

3. Set the **Prefix** or **Suffix**. Both are optional (eg. “REQ”).
4. Type in the value for your first requirement in the **Counter** box (eg. “0002” or “99”).
5. Check the **Active** checkbox to enable the auto naming.
6. Click **Save** to save your changes.

The first requirement will take the value entered into the counter box and Enterprise Architect will then increase the value by one for every new requirement (eg. “0003” or “100”).

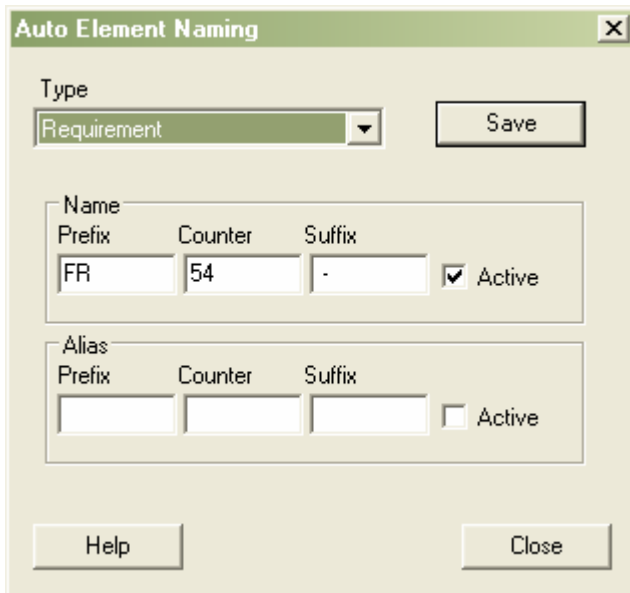


Figure 15: The auto element naming window showing how to set the counters for a requirement

### Common Uses

- If your industry, organization, company, project, or team has naming standards, this mechanism can be used to ensure elements are named appropriately.

## Drag and Drop Realizations

A fundamental aspect of the management of requirements is the ability to trace the parts of the system that implement, or realize, a particular requirement. A quick method of generating a realization link is to drag a requirement element from the project browser over an element in a diagram, which is to be the implementing element. EA will interpret this as a request to create the realization link and do so automatically.

### Common Uses

- When a project member starts to create new elements in the analysis or design disciplines, it is useful to use this technique to ensure the new elements have a purpose in the model, and are being built because they realize some requirement.

## Importing External Requirements Using the CSV Import

It is not uncommon for requirements to be initially entered into a document or a spreadsheet using some standard text formatting. Enterprise Architect provides a mechanism for importing text with a fixed structure. The simplest method is to import these text files into a spreadsheet and export this text as a CSV (“Comma Separated Values”), or tab delimited format file.

The following is a simple example of a spreadsheet containing a set of requirements to be imported into Enterprise Architect.

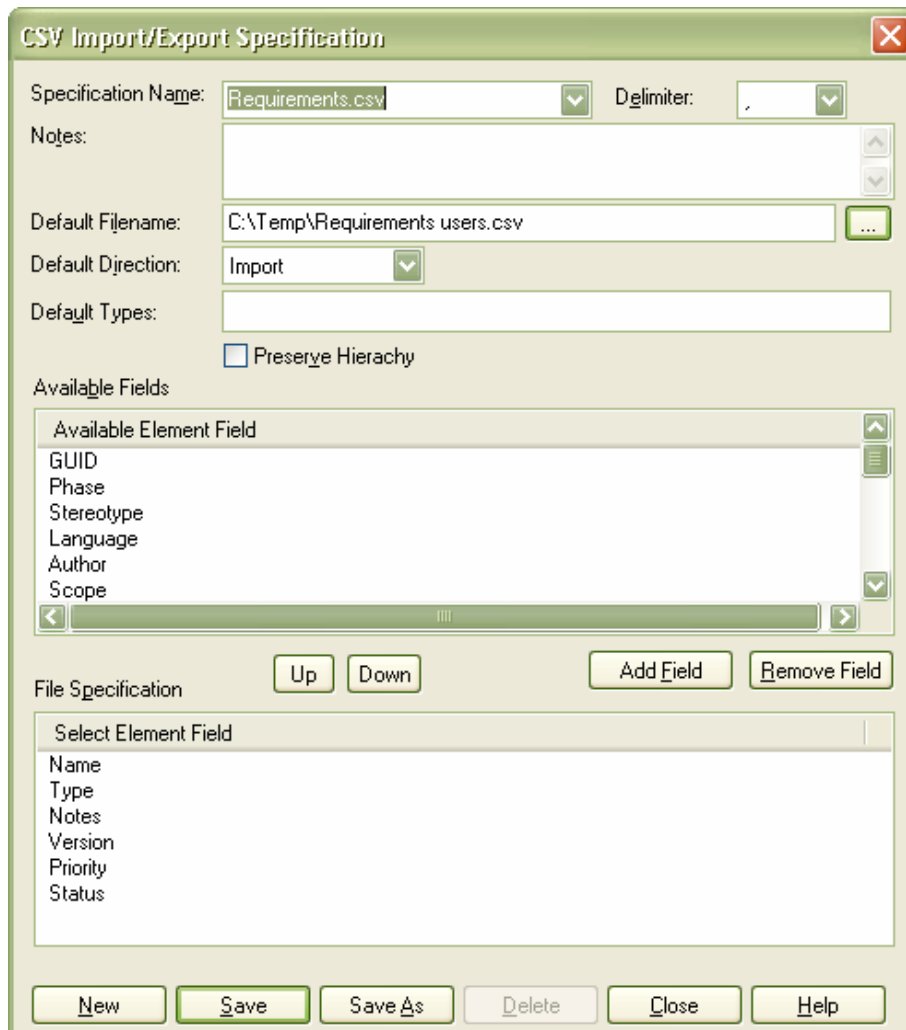
req.xls								
	A	B	C	D	E	F	G	H
1	Name	Type	Notes	Phase	Version	Status	Priority	Difficulty
2	FR27 - Retrieval of histo	Requirement	The system must be able to	1	1	Approved	Medium	Medium
3	FR28 - Transmission of	Requirement		1	1	Mandatory	Medium	Medium
4	FR29 - The staff must n	Requirement	The system must use standard u	1	1	Proposed	Medium	Medium
5	FR30 - The interface mu	Requirement	Critical to the success of the app	1	1	Proposed	Medium	Medium
6	FR44 - The interface sh	Requirement	An important aspect of the propo:	1	1	Proposed	Medium	Medium
7	FR5 - 2000 hours mean	Requirement	The Mean time between failure (f	1	1	Approved	Medium	High
8	FR6 - Must be recoverat	Requirement	In the event of software or hardwa	1	1	Proposed	Medium	Medium
9	FR7 - 99.999% accurac	Requirement	The system accuracy defines	1	1	Validated	High	High
10	FR8 - 99.999% precisio	Requirement	The precision	1	1	Implement	Medium	Medium
11	FR17 - All messages m	Requirement		1	1	Proposed	Medium	Medium
12	FR34 - Physical storage	Requirement		1	1	Proposed	Medium	Medium
13	FR35 - Wherever possib	Requirement	The organization has made signif	1	1	Proposed	High	High
14	FR31 - Other transmissi	Requirement	The system should be able to ad:	1	1	Proposed	Medium	Medium
15	FR33 - Minimal or no de	Requirement		1	1	Proposed	Medium	Medium
16	FR43 - Messages kept t	Requirement	There is a legal requirement that	1	1	Proposed	Medium	High
17								

Once completed, this spreadsheet is saved as a CSV format file. For example, if you are using Excel as the spreadsheet application, you would simply select **File | Save As**, in the field – *Save as Type*: Select \*.CSV.

### Import into EA

To import the file into Enterprise Architect, you need to create a CSV import structure that corresponds to the columns in the CSV file. To do this, select from the main menu **Project | Import/Export | CSV Import/Export Specification**.

This will return the following window:



To set up a template:

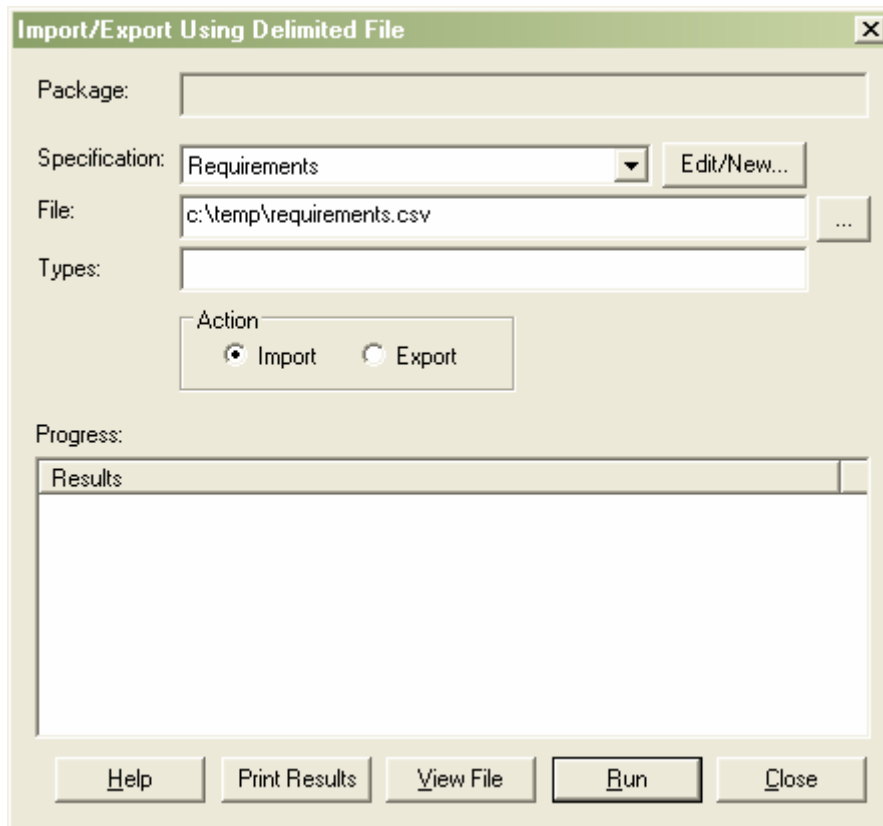
- Give it a specification name
- Define the default filename the specification will use
- Set **Default Direction** to *Import*
- Select the key fields from **Available Fields** using **Add Fields** button to place them in the **File Specification** group.

**Note:** The order of the elements in the file specification must match the order of the columns in the spreadsheet.

Assuming the spreadsheet has been saved to a CSV format, you can now import it into Enterprise Architect. It is recommended to first create a new package in Enterprise Architect that will contain the imported elements.



To start the import process, select from the main menu **Project | Import/Export | CSV Import/Export**. This invokes the following window ready for you to enter information needed to perform the import.

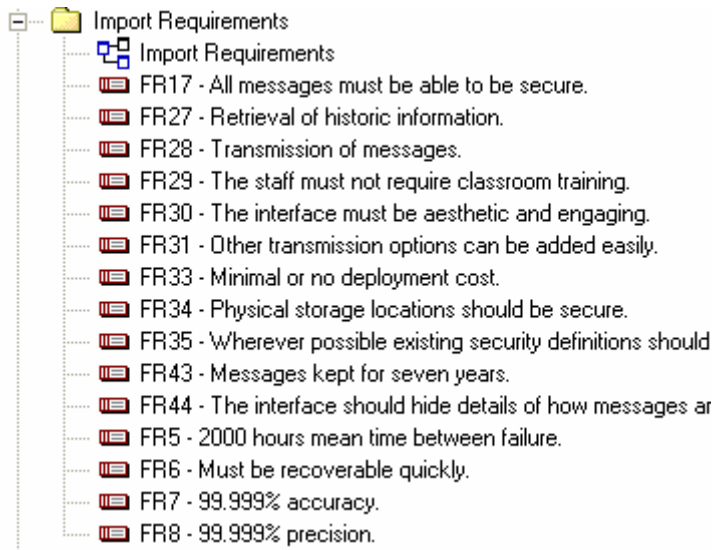


To run the import, you need to fill in the following fields:

- **Specification:** The CSV format that was defined above should now be selectable from this drop-down field.
- **File:** Insert the file location of the CSV file created from the spreadsheet.
- **Action:** Set the action to *import*.

Select **Run** to start the import process.

The data imported will be placed in the currently selected package. Below is a view of the data imported (via CSV) from the spreadsheet above.



Once imported, the requirements can then be placed in different packages by dragging the elements in the project view to their correct package.

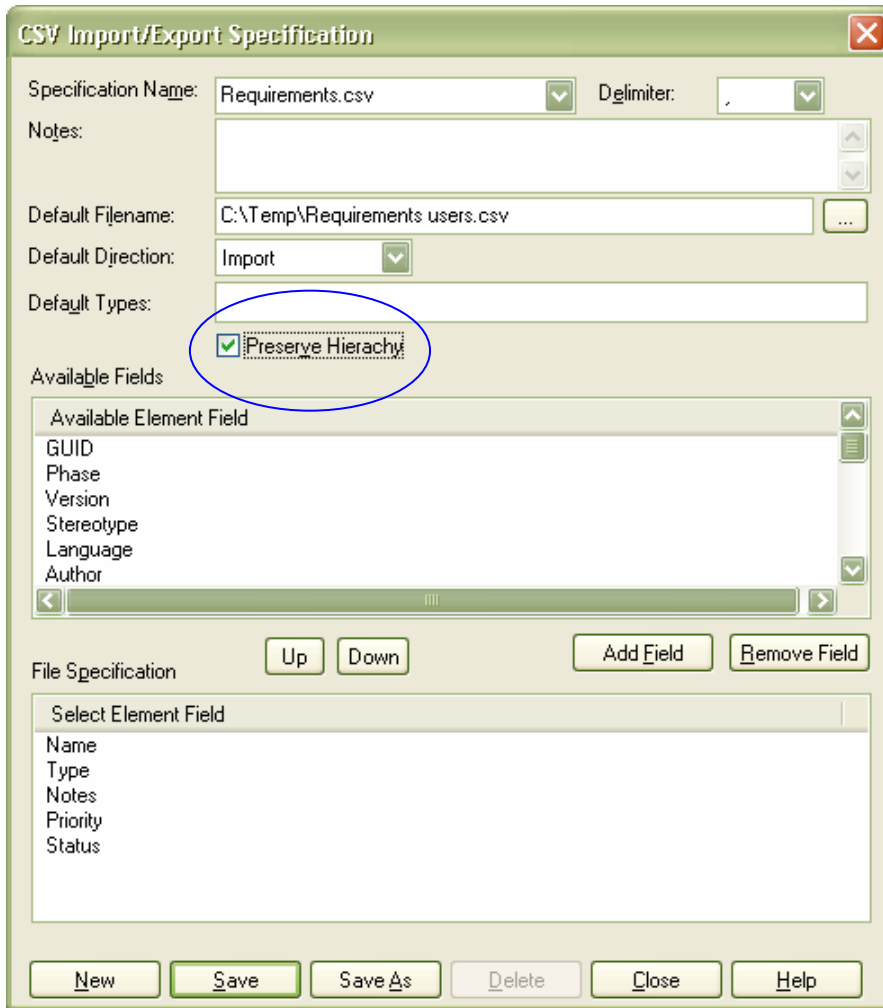
## Import a Hierarchy of Requirements

The CSV import supports importing packages and elements that are in a hierarchical form. To do this you need set up the following two fields in the CSV file:

- CSV\_Key – a unique Identifier for the Package/Element.
- CSV\_Package\_Key - the Identifier of the parent-element. This is used for arranging the parent-child relationship.

**Note:** These fields must be the last two columns in the above order.

To import a hierarchy, in the CSV specification you need to tick: [x] Preserve hierarchy as shown below:

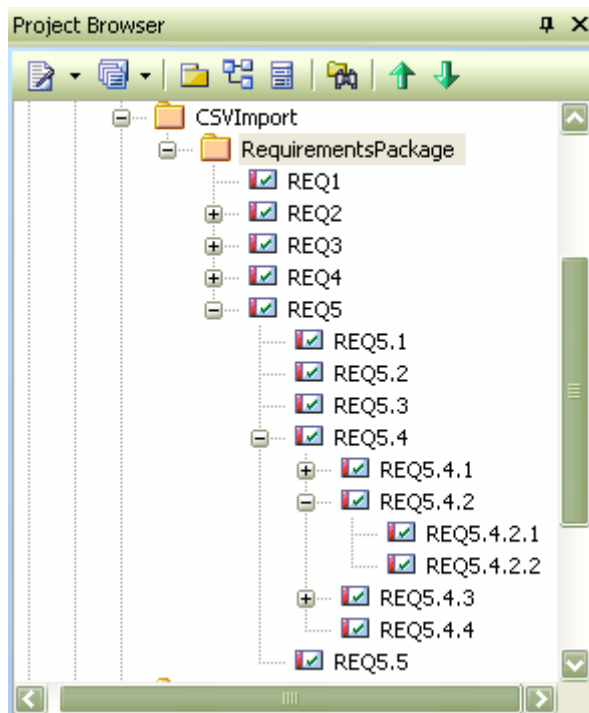


The table below contains sample data that reflects the text-formatting used with the above specification:

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
REQ5	Requirement	Notes on REQ5	Med	Approved	REQ5	Package1
REQ5.1	Requirement	Notes on REQ5.1	High	Approved	REQ5.1	REQ5
REQ5.2	Requirement	Notes on REQ5.2	High	Approved	REQ5.2	REQ5

REQ5.3	Requirement	Notes on REQ5.3	High	Approved	REQ5.3	REQ5
REQ5.4	Requirement	Notes on REQ5.4	High	Approved	REQ5.4	REQ5
REQ5.4.1	Requirement	Notes on REQ5.4.1	Med	Approved	REQ5.41	REQ5.4
REQ5.4.1.1	Requirement	Notes on REQ5.4.1.1	Med	Approved	REQ5.4.1.1	REQ5.41
REQ5.4.2	Requirement	Notes on REQ5.4.2	Med	Approved	REQ5.4-2	REQ5.4
REQ5.4.2.1	Requirement	Notes on REQ5.4.2.1	Med	Approved	REQ5.4.2.1	REQ5.4-2
REQ5.4.2.2	Requirement	Notes on REQ5.4.2.2	Med	Approved	REQ5.4.2.2	REQ5.4-2
REQ5.4.3	Requirement	Notes on REQ5.4.3	Med	Approved	REQ5.4.3	REQ5.4
REQ5.4.3.1	Requirement	Notes on REQ5.4.3.1	Med	Approved	REQ5.4.3.1	REQ5.4.3
REQ5.4.3.2	Requirement	Notes on REQ5.4.3.2	Med	Approved	REQ5.4.3.2	REQ5.4.3
REQ5.4.3.3	Requirement	Notes on REQ5.4.3.3	Med	Approved	REQ5.4.3.3	REQ5.4.3
REQ5.4.4	Requirement	Notes on REQ5.4.4	Med	Approved	REQ5.4.4	REQ5.4
REQ5.5	Requirement	Notes on REQ5.5	High	Approved	REQ5.5	REQ5

Below is a Project Browser view of the hierarchy imported using the above CSV file:



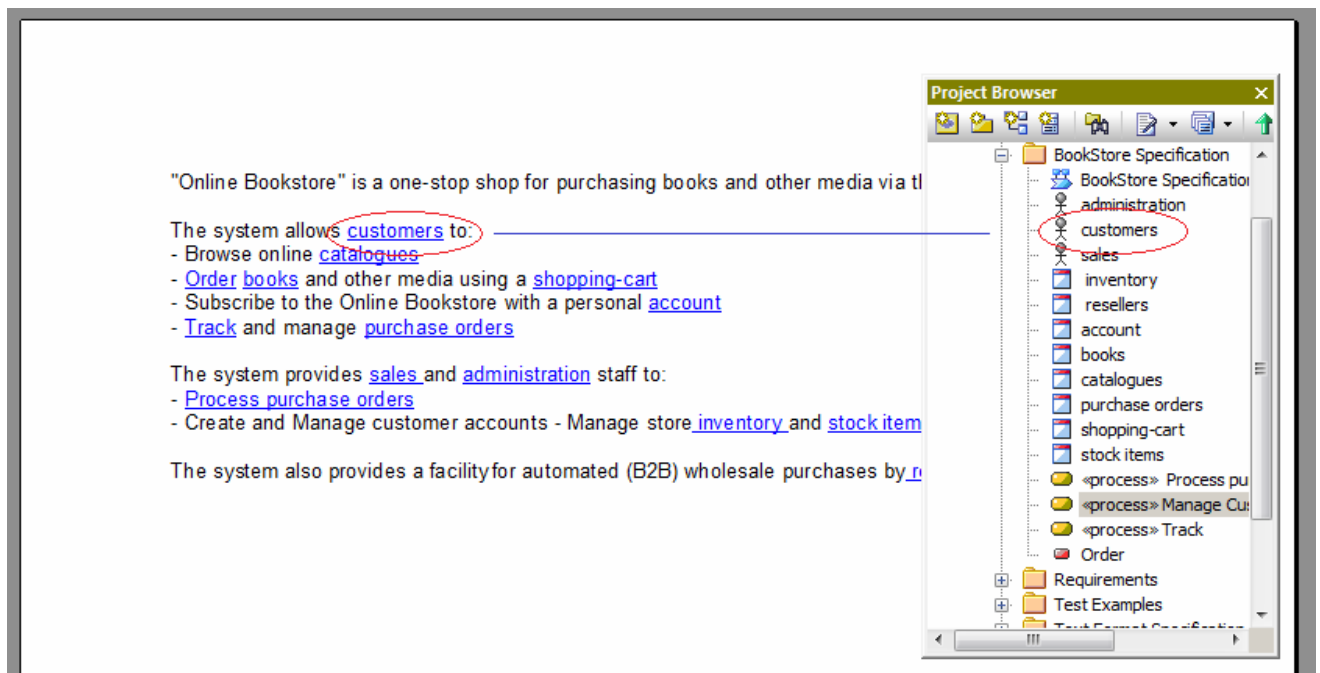
**Tip:** The above text-table can be copied to a spreadsheet and used as a starter for a hierarchical requirements document. The final spreadsheet needs to be saved in .csv format ready for import into Enterprise Architect.

## Document Based Requirement Creation

In the early stages of analyses, requirements may be defined in a text document. You can use this document to kick-start the process of formally modelling requirements. For example, you can import the document as a UML 'Document Artifact' or as the 'Linked Document' of an element. You can then select appropriate keywords in the Document Artifact (or Linked Document) and create analysis or

requirement elements directly, using the context menu. In this way, you also achieve traceability between the original requirements document and the model.

This process of dynamically creating elements from text establishes a hyperlink from the entry in the document to the corresponding element in the model hierarchy. The hyperlink allows you to trace directly from the text-based definition to the associated meta-data (such as detailed notes, constraints and status) that are subsequently defined in the semantic model. An example is given below.



To create an Element from the text simply select a section of text, right-click and in the context menu, select: **Create | New** – then select the from element type options.

## An Introduction to Use Cases in Enterprise Architect

Use cases are used to model single tasks a user of the system may perform. They give a slightly more complex definition of the process involved in a system that conforms to the requirements laid down.

Enterprise Architect allows you to draw use case diagrams, and to specify the use case in a number of different ways. In addition to the introduction here in this section, Enterprise Architect contains use case related features such as activity, sequence and state diagrams.

### Use Case Diagrams

Use case diagrams describe how a user of the proposed system will interact with this system to perform a discrete unit of work. It describes a single interaction over time that has meaning for the end user.

Use cases typically have requirement, constraint and scenario definitions associated with them. These describe the essential features and rules under which the use case will operate. Below is a simple example of a use case for an email-based contact and address book.

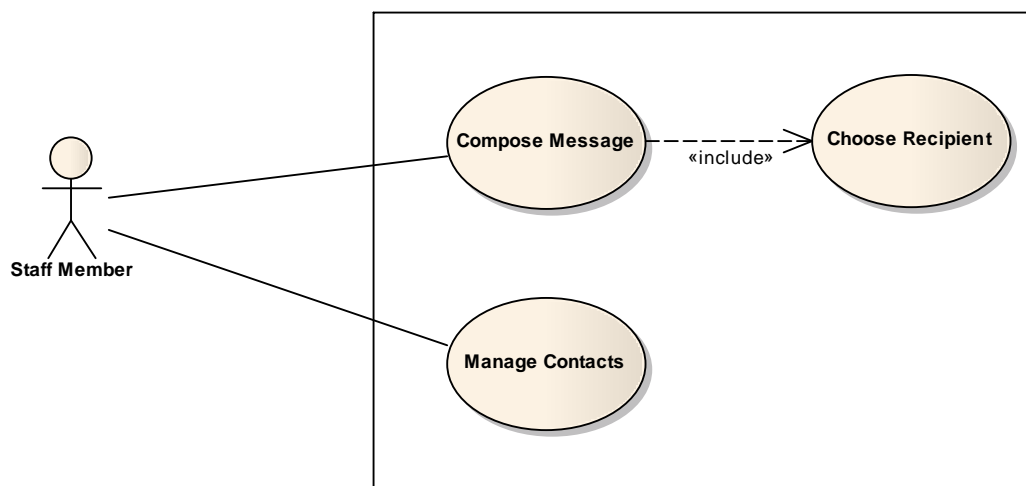


Figure 16: Use case diagram showing actors and use cases.

### How to Create a Use Case Diagram

Enterprise Architect provides a use case model you can use. You can either include it when creating a new project, or **right-click the model in the Project View** and select **Add Model from Pattern**. This will provide you with further information on use cases.

Once you have added the use case model to your project, navigate to the use case model diagram and double left-click to open it.

Open the **Use Case palette** from the **Toolbox** on the left of the Enterprise Architect interface. The elements listed here include actor and use case. These elements can be dragged onto the diagram in the same way as requirement elements. Relationships can also be defined in all the same way as between requirements.

### Common Uses

- To define the scope of the system;

- To define the people and other systems that will use the system;
- To document the way the business process is performed; and
- To provide the basis for the user documentation, help system or manuals.

### Tips and Tricks

- **Consider using a classifier style representation of a use case for all system actors.** Newcomers are often confused (not unreasonably) by a stick person representing something non-human, like another system.
- **Consider putting the actors (primary or active) that initiate a use case on one side of the diagram,** and the actors (secondary or passive) that don't initiate the use case, but play a part in the use case on the other side.

## Defining Scenarios

The use case property sheet allows you to specify attributes that apply to the use case as well as detailing the scenarios. Double clicking on a use case element allows you to define the specifics of the use case.

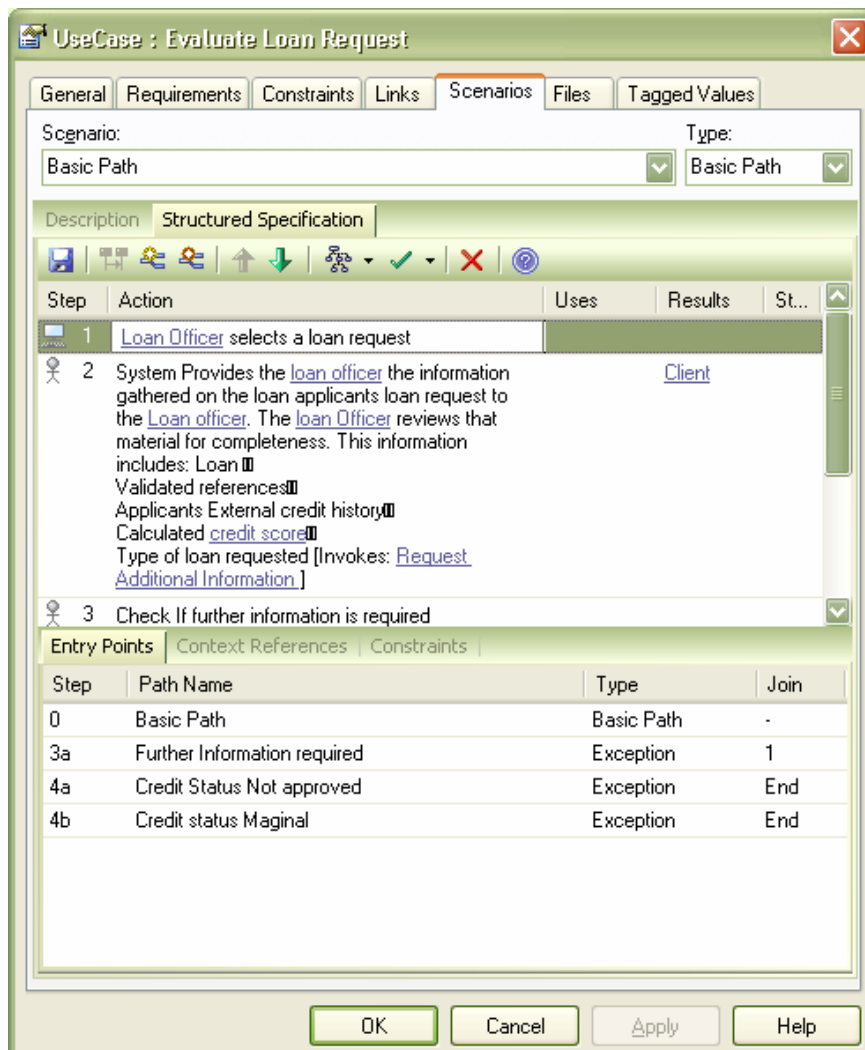


Figure 17: The property sheet for a use case showing the scenarios tab

For more details on using the Structured Scenarios see the help section: [The Scenarios & Requirements Window](#)

### *Linking with Requirements*

Using the realize relationship, you can define which use cases are implementing the requirements. See [Tracking and Relating Requirements](#) for more information on creating these links.

## **Additional Features of Enterprise Architect**

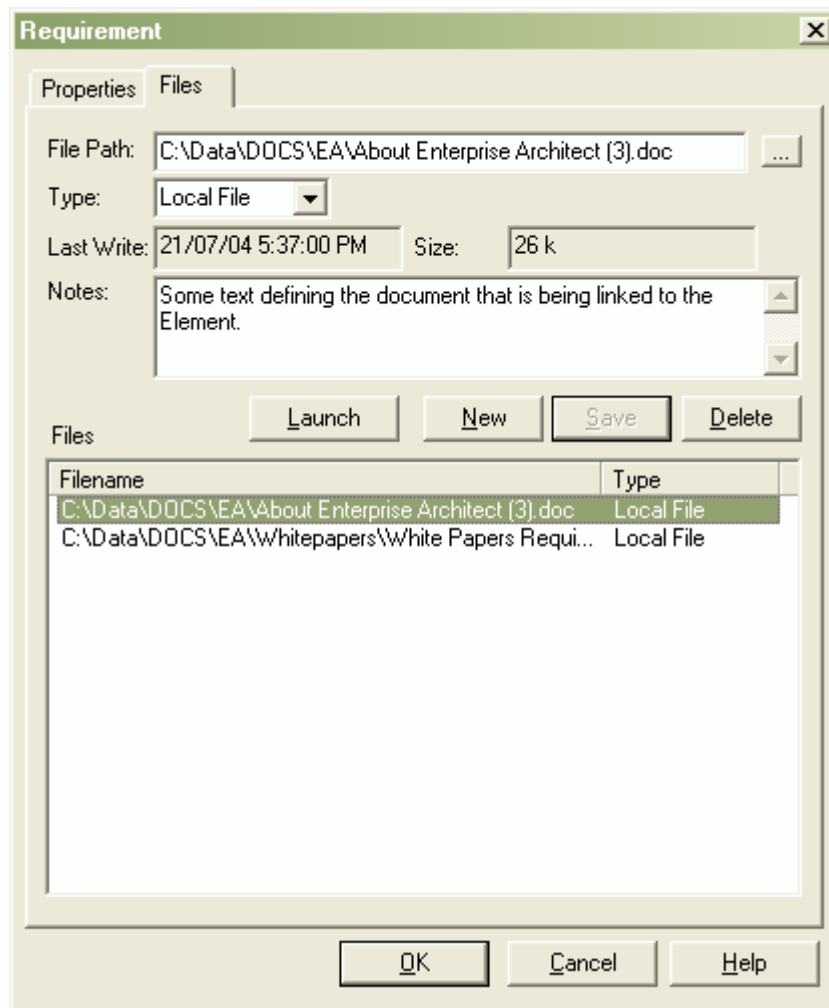
Several of the features Enterprise Architect provides are useful across any modeling you may decide to undertake.

### *Attaching Documents and Files*

The UML specification allows for textual descriptions for each element, and Enterprise Architect provides this facility. If you have documents that need to be referenced, these can be linked to the element (a requirement, a use case, etc.), using options that allow for external documents to be linked or internal RTF documents created and stored internally. The following covers these options.

- 1) Elements and the **Files** tab.  
External files can be linked to an element using the **Files** tab in the element properties window. Below is the properties dialog for a requirement with the files entries and a listing of linked documents.

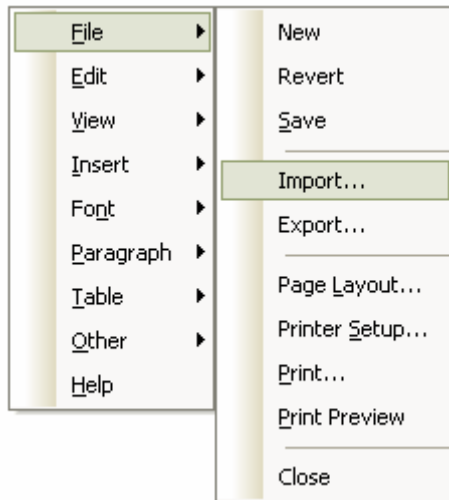




- 2) **Linked Documents.**  
Each element can have an internal RTF document linked to it. This is accessible by selecting the element, then right-clicking, and from the context menu selecting the **Linked Document** option. This will open the RTF editor for editing.
- 3) **Using a Document Artifact Element.**  
The document artifact element is available from the **Toolbox** under **Deployment**. After creating and naming this, subsequent double-click selection of the element will open the RTF editor for word processing the internal document.

**Note:** Linked documents and document artifact documents can be referenced in RTF report templates using the Sections: **Element | Linked Document** and **Package | Linked Document**.

Options two and three above allow for external documents to be imported using the RTF editor menu. This menu is accessible by using a right-click on the body of the document. The following shows the import and export options selection.



### Common Uses

- To attach a textual document that describes the requirement or use case. Organizations often require use cases to be described using text. In these situations it is beneficial to make the document available by attaching it from within Enterprise Architect.
- Formal business specifications including regulatory constraints and legal requirements may be attached as files making them available for all project members to view.

### Tips and Tricks

- Consider attaching the relevant pages of your organization's standards either as a document or as a web page on your organization's Intranet Site.

## The Glossary Function

Having a shared description of a term is important when relating new concepts to other parties involved in the system development process. The Enterprise Architect glossary allows you to enter terms and their definitions or descriptions directly into the model glossary, or when typing a new term into a notes field, it can be added to the glossary. These terms are then highlighted as glossary terms in the Notes.

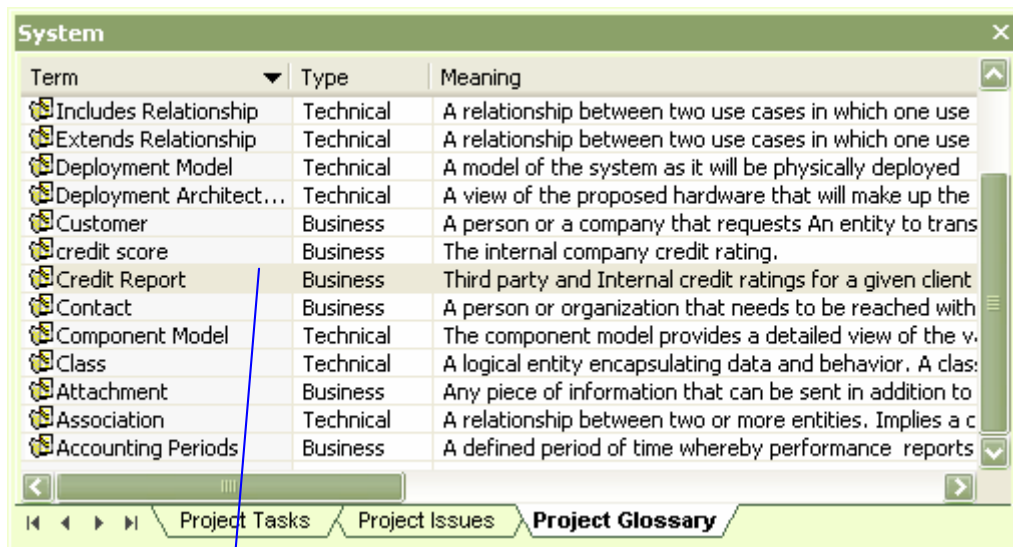


Figure 18: Glossary window showing business and technical terms and descriptions

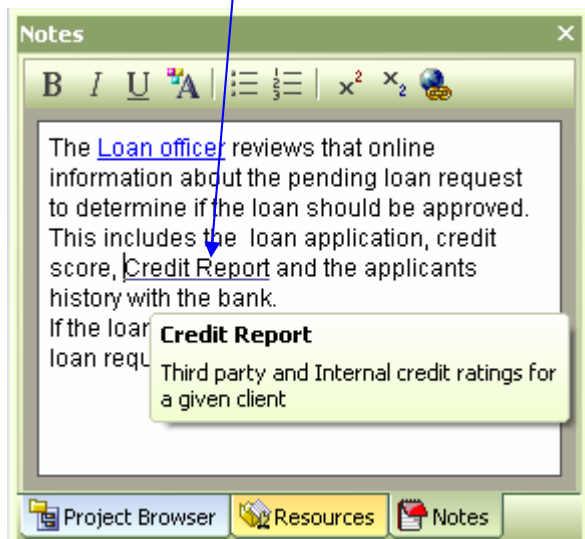


Figure 19: A selected Glossary item in Notes – with the description in the context reference window.

### Common Uses

- Provide definition of process-related terms, such as the definition of a formal requirement or a process worker.

### Tips and Tricks

- Consider reusing the glossary from a previous or related project. The common terms that relate to your domain can be included in a base project (this can be exported from one repository and imported into another repository using the option on the main menu **Tools | Export Reference data** and **Tools | Import Reference data**).
- Consider generating a report (available from the glossary window) from the glossary as an attachment or inclusion with formal documentation.
- When creating new reports using the RTF generator consider adding the glossary as an appendage to some of the key documents to be generated.

For more details see: [Project Glossary](#) & [The Notes Window](#)

## *Defining Requirement Attributes Using a Profile or a Template*

As stated above, the requirement element can be pre-defined to include a set of user-defined attributes. These are used to document user specific qualities. The additional attributes can be defined using either a *Model Template* or a *Profile Definition*.

A short comparison of these two options is as follows:

1. The template definition is the simplest to set up, however:
  - a. It alters the default requirements element; and
  - b. It is restricted to one set of modifications to the requirement definition.
2. The profile definition is more complex to set up, but allows multiple requirement definitions and retains the default requirement element.

Both of these require attributes to be defined using the tagged values definition.

**Note:** If the project contains a number of requirement types that contain different attributes, then a Profile, although more complex to set up, is better to use.

## Defining Tagged Values

With tagged values, the user can define any number of fields with a wide variety of predefined or user-defined data types.

To set up a standard tagged value, select from the main menu: [Settings](#) | [UML](#) | [Tagged Value Types](#). This will bring up the tagged values definition screen as shown below.

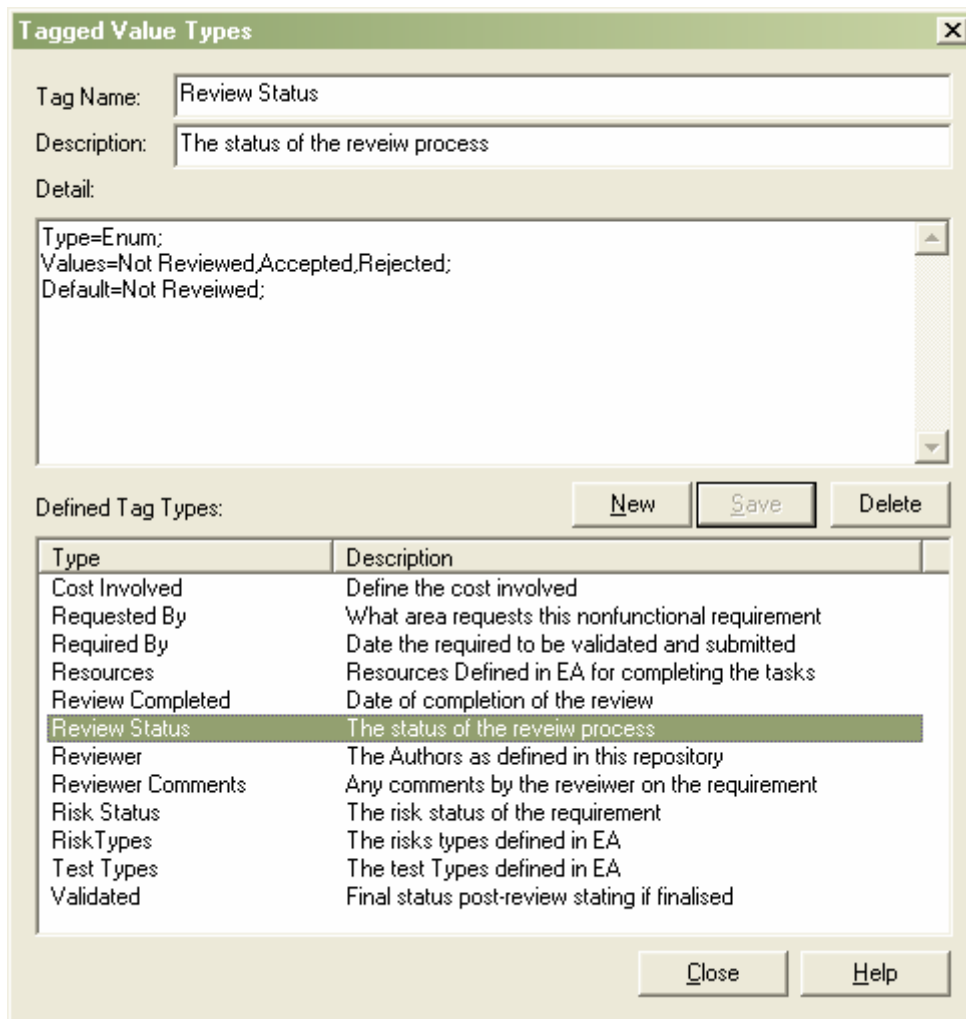


Figure 20: Tagged values definition window

In the example above, the tagged value selected, called “Review Status”, uses a predefined type to display a drop-down list of selectable options. In the detail area it contains:

Type=Enum;

Values=Not Reviewed,Accepted,Rejected;

Default=Not Reviewed;

When viewed in the tagged values window, this is presented as a drop-down option box.

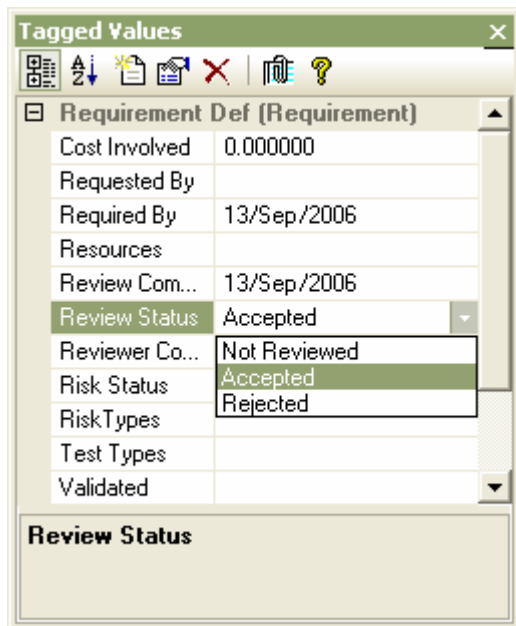


Figure 21: An element's tagged values

There are numerous standard types available, such as numeric and string types; Enumerated lists (see above), Date-Time, Boolean, Memo, etc.

For more information on setting up the standard types, and a list of types available in EA's help: use the [Index](#) tab to locate ['Predefined Structured Types'](#)

## Defining Requirements in a Model Template

EA supports the creation of a template package. This can be used to define specific settings for any element type (i.e. a requirement element). An element of a specific type can be placed in the template. Any feature settings defined on this element will dictate the feature settings whenever a new element of this type is created. This is commonly used for re-defining the properties, such as default color, default stereotype, etc.

When dealing with additional attributes for requirement elements, the template can be used to define a set of tagged values that are assigned to any new requirement element.

To create a template for the model:

1. Set up a specific package that will contain the definition element(s)
2. Create a diagram for this package
3. From the main menu, select [Settings | Template Package](#).

For more information on this topic, in EA's help, perform an [Index](#) search using the term [Templates Package](#).

## Defining Requirements Using a Profile

EA supports the creation of profiles. Profiles allow the user to define a set of extensions to standard EA elements using tagged values.

In simple terms, this is similar to the template definition above, however it also allows for:

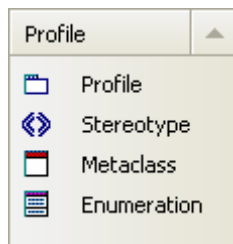
- Any number of new element types to be defined for one standard EA element type (i.e. multiple and different requirement element types);
- The standard element type to remain unchanged; and

- A toolbox group to be defined to make the new elements user-accessible.

As is the case with a template definition, it is necessary to define a set of tagged values that will be used in the profile definition. See [Defining Tagged Values](#) above.

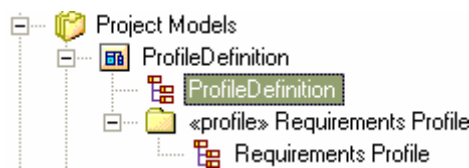
To create a profile:

1. In the project browser, set up a specific package that will contain the profile.
2. Create a diagram for the profile under this package.
3. In the **UML toolbox** open the **Profile** section.



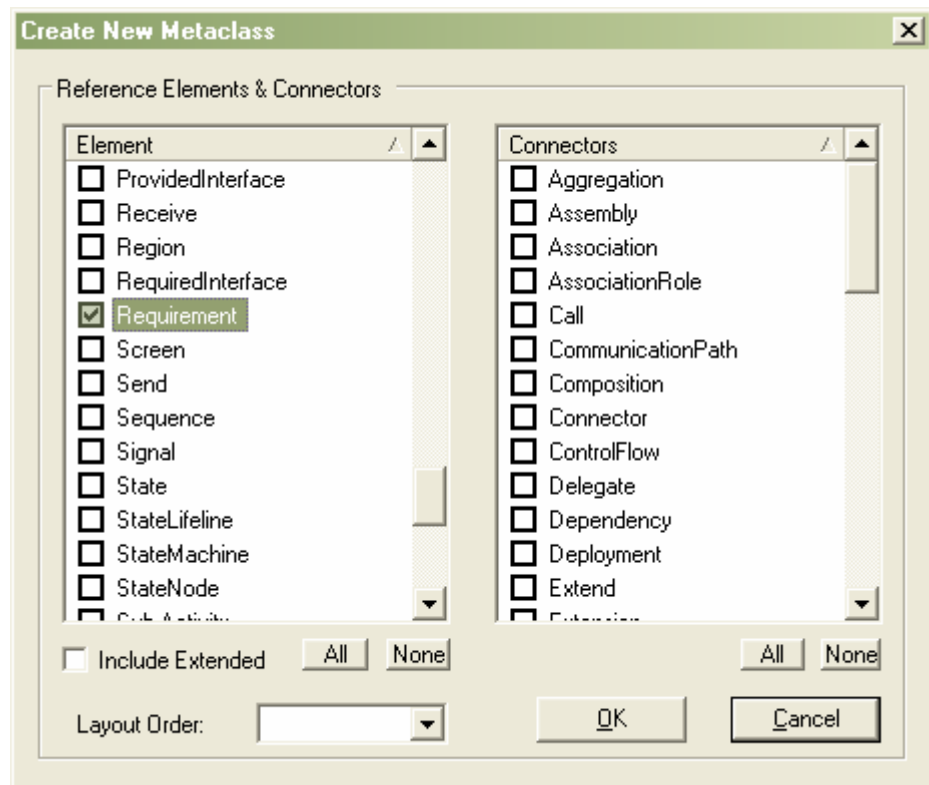
5. Drag the **Profile** element onto the diagram.

Once you have entered a name for the profile, a package will be created with the **<<profile>>** stereotype and a child diagram beneath it. This child diagram (eg. 'Requirements Profile' in the image below) will be used to add stereotypes to the profile.

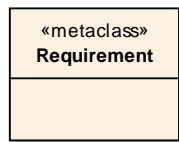


To add a new requirement definition:

1. Open the system created diagram contained in the **<<profile>>** package.
2. Create a **Metaclass** element by dragging the **Metaclass** tool from the **Profile** Toolbox. This will bring up a dialog box to select the type of element that is to be created.
3. For a requirement select the **Requirement** element type.
4. Press **OK**.



The metaclass will appear as shown below.

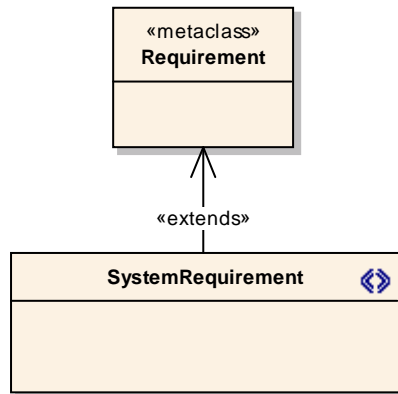


To define where the tagged values will be included in the profile:

1. Drag the **Stereotype** tool, from the **Profiles** toolbox, onto the diagram.
2. In the **Properties** dialog box, select the **Name** field. Type in your preferred name for the element in the new toolbox, i.e. 'SystemRequirement'.
3. Press **OK** in the **Properties** dialog.
4. From the **Profile** toolbox click on the **Extension** connector, select the new stereotype element 'SystemRequirement' and drag the mouse to the requirement metaclass. This should create an extension connector between these two elements.

Below is a view of the elements and connection created.

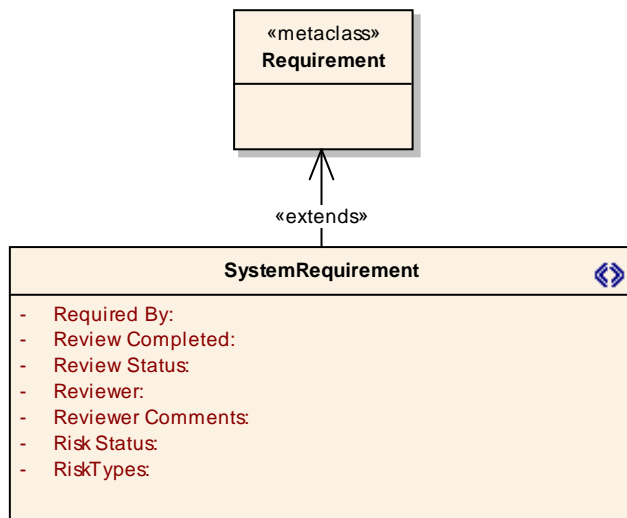




To add the tagged values:

1. Select the stereotype (i.e. 'SystemRequirement')
2. From the context menu, select **Attributes**
3. In the **Attributes** dialog box, add as a new entry in the **Name** field for each of the tagged values created above, you want included in this requirement element.

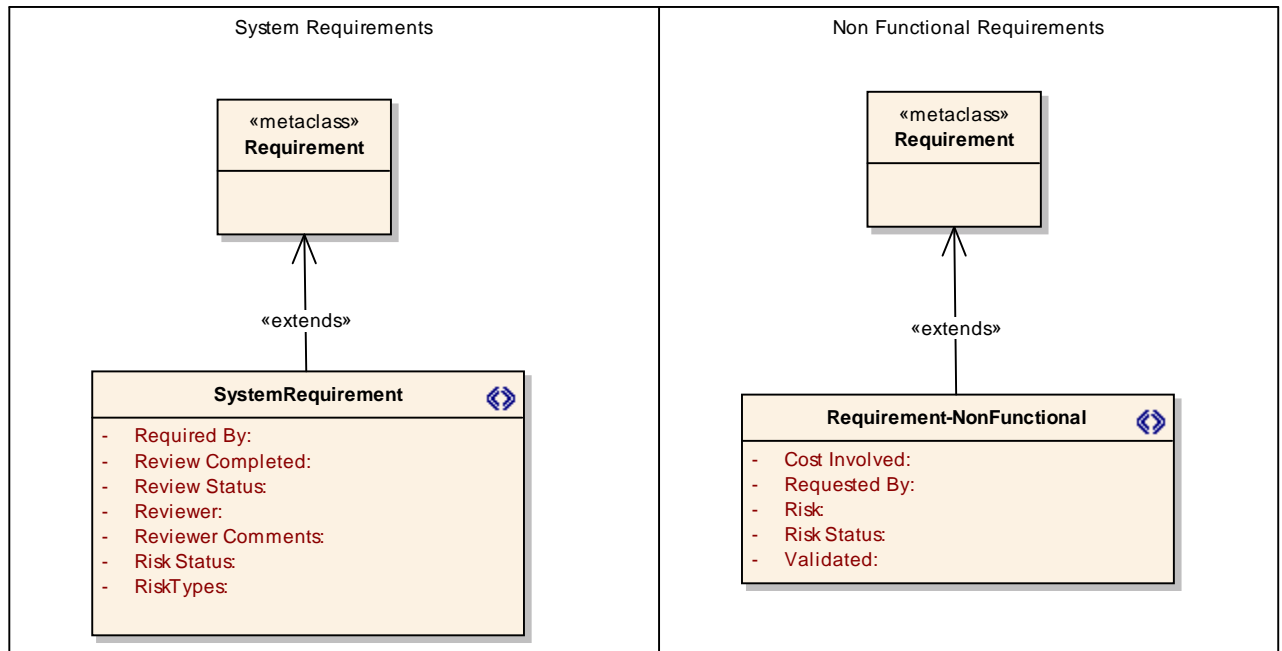
Below is an example of the above 'SystemRequirement' populated with some of the tagged values defined in Figure 20: Tagged values definition window.



**Note:** In the case of creating simple string or Boolean tagged values, these do not need to be defined in the tagged values definition, but can be entered directly in the attributes of the 'SystemRequirement' element.

Any number of these definitions can be set up in one profile. Each stereotype element would need to have a unique name.

Below is an example of the definition of two different requirements.



To set up the new requirements to be viewed in the Toolbox:

1. Select the **<profile>** package.
2. Right-click, and from the context menu, select **Save Package to UML Profile**.
3. Set the filename to save the XMI file.
4. Select **Save**.
5. Open the **Resources** view.
6. From the resources tree, select **UML Profiles**.
7. Right-click and from the context menu, select **Import Profile**.

A new toolbox with the name of your profile package will be added to **Toolbox | More Tools**. Now you can create requirement elements in any package by dragging these from the toolbox.