Requirements Management with Enterprise Architect

By Sparx Systems
www.sparxsystems.com

Trademarks

Table of Contents

Introduction

Enterprise Architect integrates 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.

Representing a requirement as a UML element helps you to trace requirements 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.

Requirements Management can involve a number of different steps ranging from the broad definition of the process your organization will use, through to implementation of these requirements within your model. Requirements Management processes differ from one organization to another, but can include any of the following:

- Documenting the process used for Requirements Management
- Inputting requirements (manually or imported)
- Tracing a requirement through to implementation
- Change Management
- Team Interaction and Review
- Project Management
- Testing
- Documentation

Enterprise Architect offers a range of tools that you can quickly use for overall management of your requirements for any of the above processes.

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 introduced a new way to describe functional requirements, the Use Case. While this was 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 of different use case styles and interpretations. In this paper we will discuss many of these issues and how to use Enterprise Architect to create and

manage requirements, in a text format style, but using a UML profile specific for Requirements Management.

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 Enterprise Architect can be used to define requirement elements, link requirements to model elements that implement them, structure requirements into a hierarchy and report on requirements.

Before we get started, 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. To open the example model, from the **Start Page**, select the 

icon.

Requirements modeling

Entry of requirements into the model is only one stage in the process of integrating your requirements with other aspects of the model. After requirements entry, there are a variety of facilities for working with requirements and specifications. Figure 1 gives an outline of the key functionality in Enterprise Architect useful in Requirements Management.

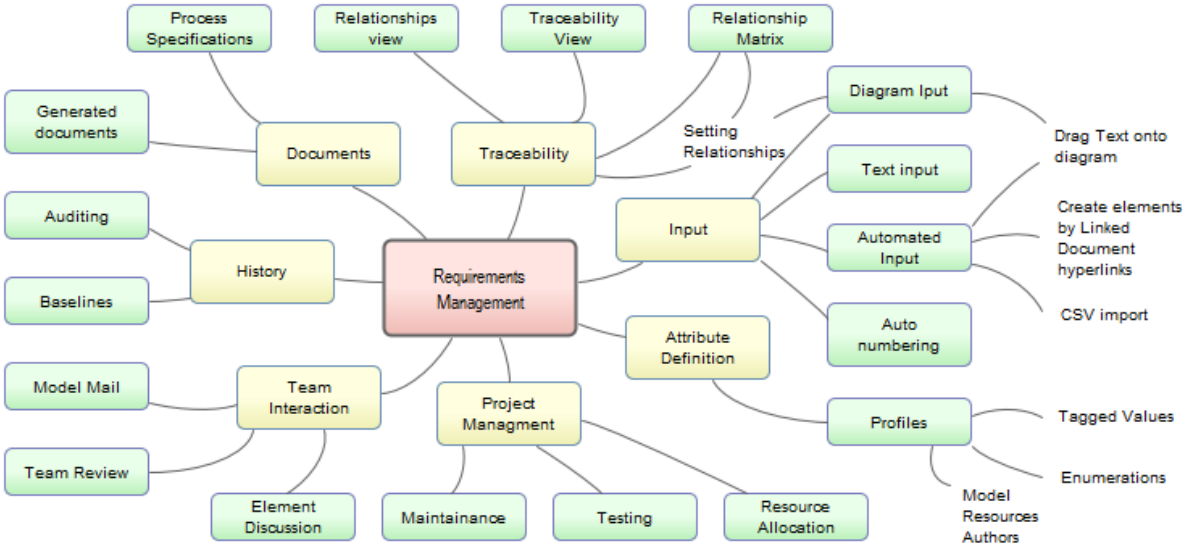


Figure 1: Outline of the functionality that can be applied in Requirements Management

In this paper you will be introduced to the general process of Requirements Management and the tools available in Enterprise Architect to implement your process. The key points covered include:

- Defining and documenting your Requirements Management process
- Setting the Attributes that your requirements need to store
- Inputting requirements (manual and automated)
- Relating your requirements to aspects of the model

- Tracing these requirements in the model
- Maintaining a history of changes to your requirements
- Team based interaction for input and reconciliation of requirements
- Documenting requirements

This paper will provide a brief introduction to each of these aspects along with tips for the best use of the features for implementing them.

Defining your Requirements Management process

As with any modeling endeavor, a variety of methodologies can be employed. With this diversity of possible methods it is good practice to document the methodology that needs to be applied in your Requirements Management.

Depending on your organizational background you might already have documented a process for defining the requirements for a new system. If so, you can quickly import this documentation into your model. Otherwise you start with a template supplied with Enterprise Architect. For more details on creating review documents see the Create a Review Document Help topic. [DOB: yet to be posted on web]

Part of this definition should be an overview of the extra fields (Attributes) that you intend to use in your Requirements Management.

Setting the Attributes for your Requirements

Depending on the system under development and the organization there can be a variety of Attributes that need to be recorded against each requirement. Enterprise Architect supports user-defined fields. These are called Tagged Values, which support a variety of formats, ranging from simple text and date values through to user-defined drop-down lists. Tagged Values can be used on a one-off basis or defined to be automatically included on creating a new element. The details on the definition of these user-defined fields will be covered in the section Defining Requirement Attributes using a Profile.

Requirements specification input

When developing the preliminary specifications for a project, there are three common methods employed:

- Text-based input of specifications
- UML diagram based requirements modeling
- Automated import of requirements from external sources

Enterprise Architect provides an integrated means of defining and working with specifications using all of these methods interchangeably. However, as an introduction we will give separate details on the use of each of them.

Specification Manager

The **Specification Manager** is a tool for users more familiar with a text-based means of creating and reviewing requirements. These users may include business professionals and managers who might not have expertise in model development.

What the Specification Manager provides is a means to enter and edit entries in a simple semi-tabulated text form. It is also an interactive reporting tool that is capable of indicating what other metadata is associated with requirements, and launching dedicated editors for such metadata. For example, you can inspect a requirement entry and instantly see whether it has associated test cases. If so, you can simply click an icon beside the requirement, which will invoke Enterprise Architect's Test Management window – ready for you to view and edit those test cases.

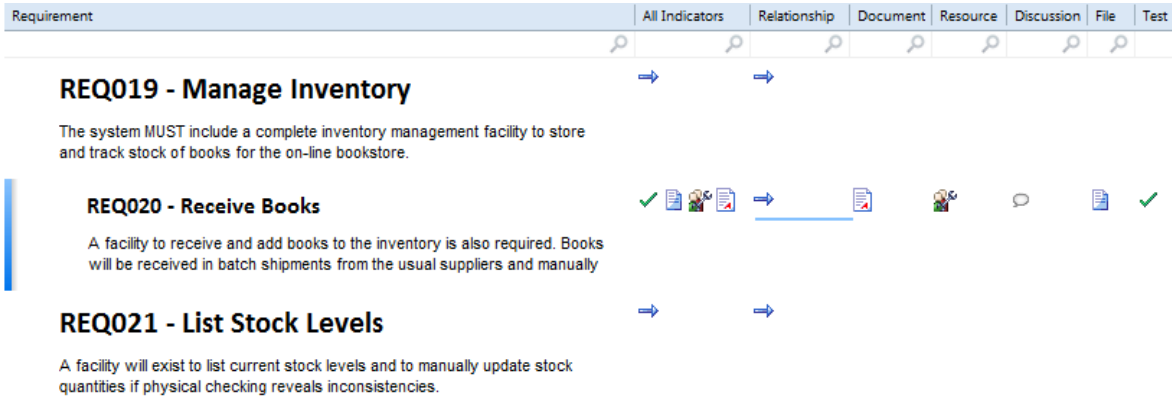


Figure 2: Specification Manager view of requirements

Adding entries

Each entry in the Specification Manager represents a model element in the Enterprise Architect project. The example entries shown in Figure 2 are Requirement Elements. New entries (elements) are added via the New Element icon or using Ctrl+N or by right-clicking on the diagram and selecting Add New Element or Add New Child. The Specification Manager therefore makes is easy and intuitive to add new requirements to your system specification.

There are a number of alternative methods for importing requirements, which will be covered in the topics under Importing External Requirements.

Nesting entries

Any new entry can be made a child or parent of another element by dragging the entry above or below an existing entry in the Project Browser.

Setting views

To give a clear and simple view of the resources associated with each specification, the Specification Manager indicates their use with icons. These icons provide a quick reference to related details like Traceability, Project Management and Change management.

Each resource-type is available as a column selectable from the **Field Chooser** dialog. For details on using these see the **Indicator Columns** Help topic.

Figure 3 is a text-based example with the **Tagged Value** window, the **Traceability** window and the **Element Discussion** window open:

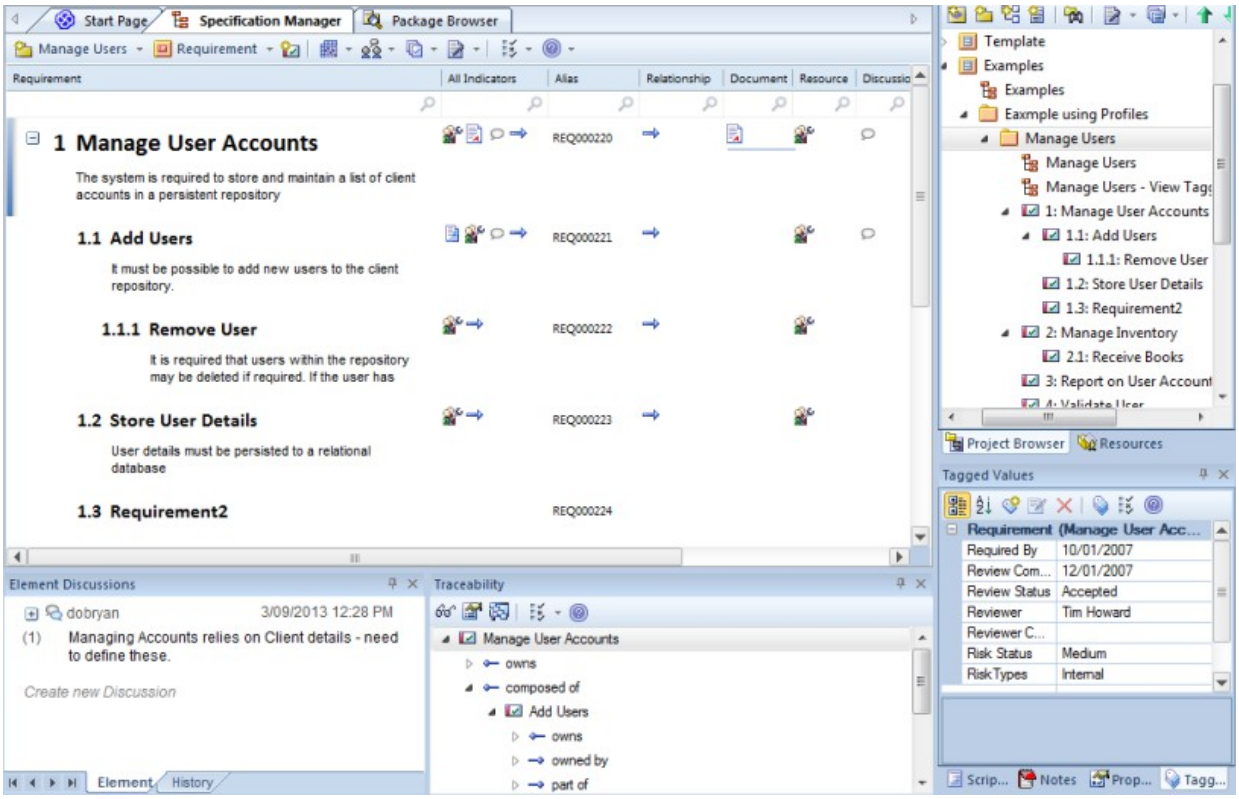


Figure 3: An alternative visual layout

Some common features to use in conjunction with the Specification Manager include:

- Tagged Values view
- Relationships view
- Traceability view
- Discussions view
- Project Management Resource view

Tips and tricks

- The Project Browser can be set to hide the Stereotype (eg. *<<Functional>>)* using: Tools | Options General | [] Show Stereotype. Figure 3 shows this option turned off.

Requirements modeling

For a more formal diagram-based representation, requirements can be shown diagrammatically with their relationships (see Error: Reference source not found). The core information behind any one requirement is defined in the properties section (see Figure 5), user-defined "Attributes" can be created using Tagged Values and Profiles (see Predefining Tagged Value Types for Requirements).

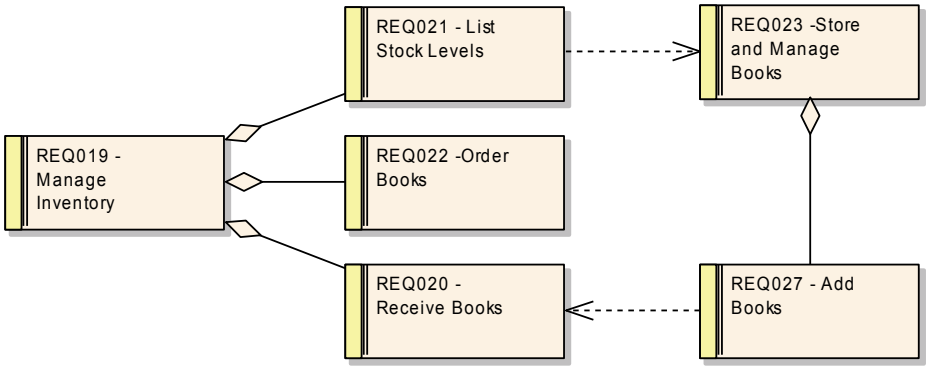


Figure 4: Custom diagram showing the requirement element

Creating a requirement element

There are numerous ways to create requirements in a diagram. The key methods are:

- Creating a new entry in the Specification Manager
- Dragging an item from the Requirements Toolbox onto a diagram
- Dragging text from an external application onto a diagram

For other options and more details on using these see the Create Requirements Help topic.

Tips and tricks

- The requirement element's 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 the Auto-Naming Elements Help topic). The auto-generated numbering can be placed in the Element's Name field or the Alias field.

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. Figure 5 shows an example of the properties for a requirement.

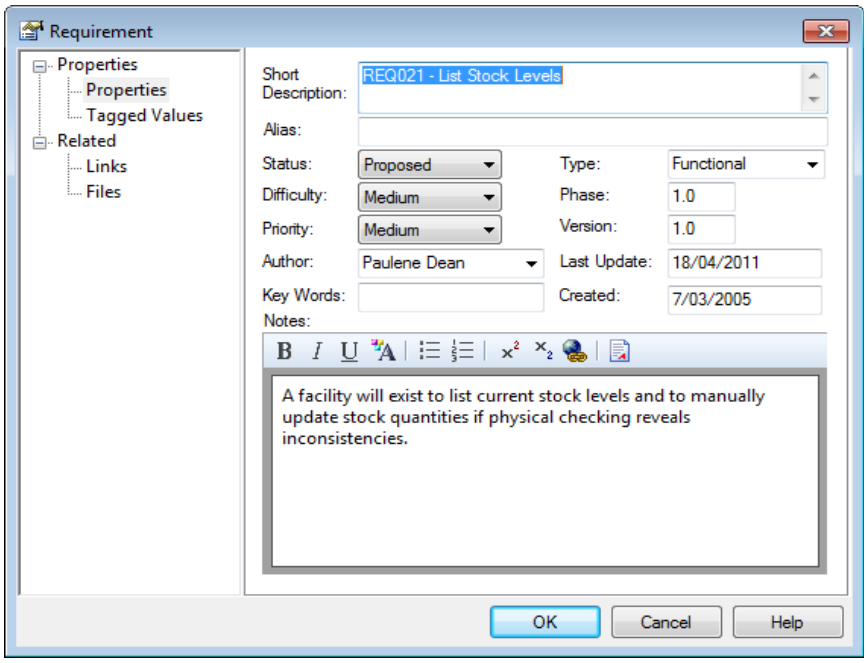


Figure 5: External requirements properties

For a more permanent docked view of the element's properties you can have open the **Element Properties** view (Alt+1) and the Notes View (Ctrl+Shift+1).

Adding Custom 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 for a specific element, or predefined to be added to all new requirement elements.

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

See Figure 6 for a diagram showing a one-off addition of a Tagged Value.

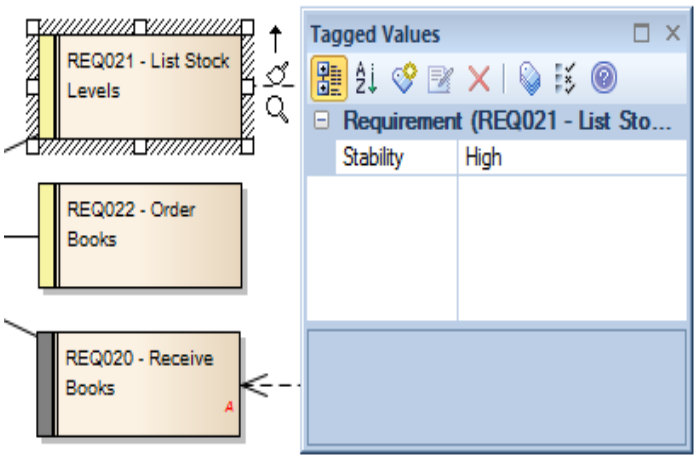


Figure 6: Requirements Tagged Value dialog allowing the assignment of Attributes

If you use Tagged Values often, consider leaving the window open and docked.

Predefining Tagged Value types for requirements

Elements in Enterprise Architect can have an extended set of Attributes defined, that are automatically created with each new element. This set is defined using a *UML Profile*. See Figure 7 for an example of an element using a predefined set of Tagged Values for a project's requirement elements.

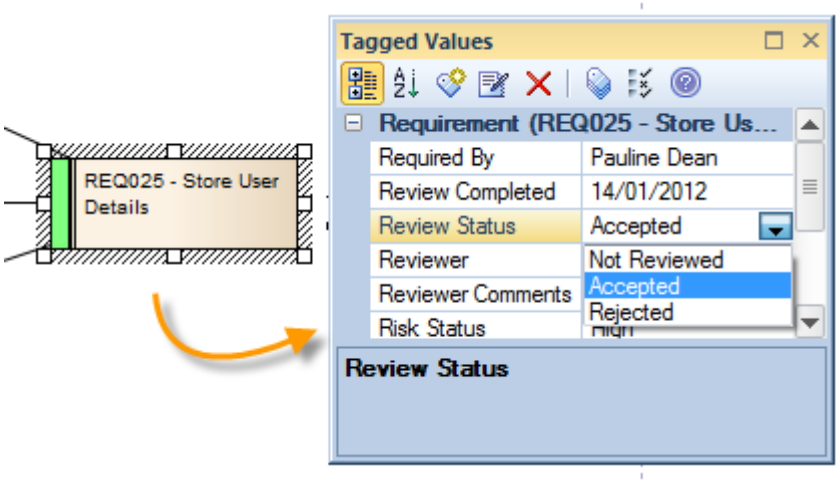


Figure 7: Using predefined tagged values

The predefined Tagged Value types can include a number of standard formats, such as date/time, calendar view and drop-down lists.

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 7 viewed in this mode.

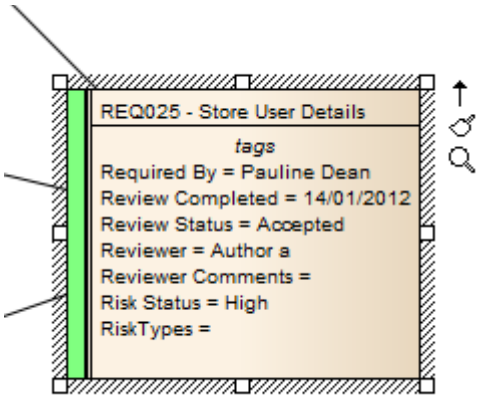


Figure 8: Tagged Values visible on elements

For more information on extending requirement Attributes using Tagged Values see: Defining Requirement Attributes using a Profile.

Auto Element naming

If your industry, organization, or project team has naming standards that include numbering, Enterprise Architect provides two mechanisms that can be used to help you name elements appropriately. You can use either:

- List Numbering, or
- Auto Naming

Figure 9 shows examples of List numbering (left circle) and Auto-numbering (right circle).

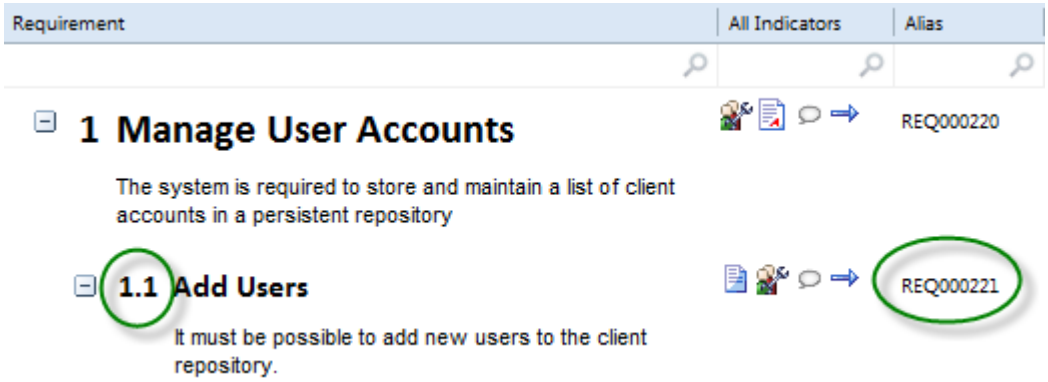


Figure 9: Examples of List Numbering (left) and Auto-numbering (right).

The following two sections explain the different advantages of each of these mechanisms.

List Numbering

List Numbering numbers the element in a 1.1.1 format based on the element's position in the tree. It is an impermanent system-based numbering, so any movement of the element in the tree will update the numbering according to the element's new position. List Numbering can be used in the different views, such as the Specification Manager, and can be reported.

Note: 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).

Figure 10 is an example an element hierarchy viewed from the Project Browser with Level Numbering set on.

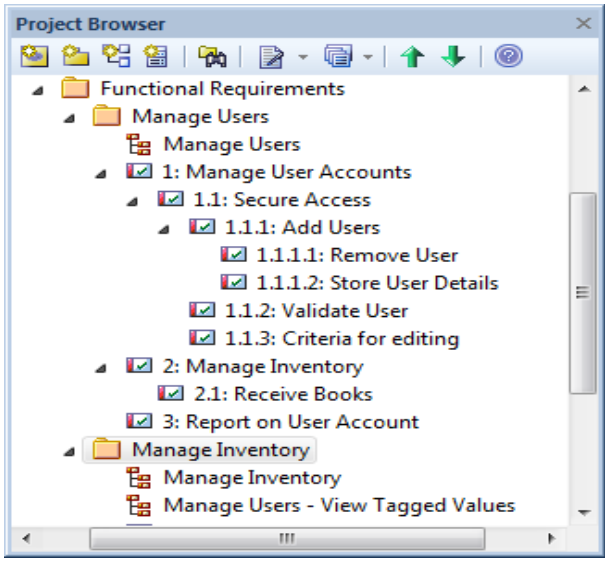


Figure 10: An Element Hierarchy with Level Numbering

To enable this option:

- Select a package in the Project Browser
- Right-click and from the context menu select: Turn on Level Numbering

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

Auto Naming

With Auto Naming, you can configure Enterprise Architect to automatically name and number requirements as they are created. It is more permanent, but can be updated. Auto Naming is particularly useful with requirements as they often require a unique reference for external checking.

Figure 11 is an example of configuring an auto name counter.

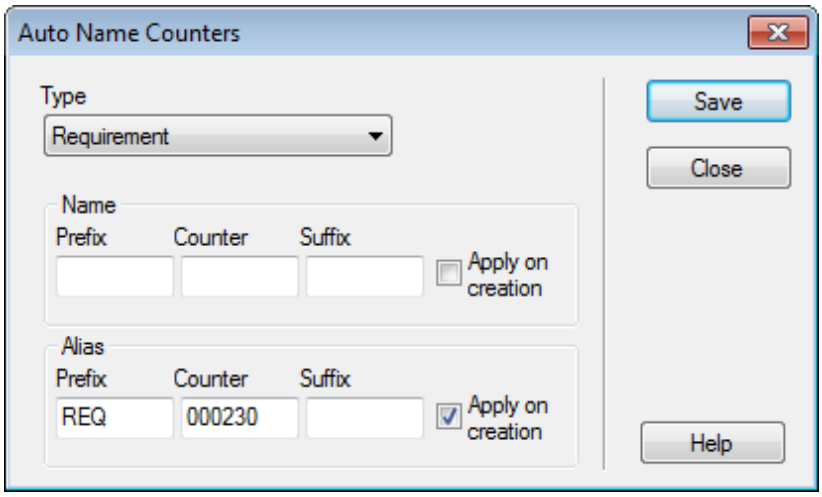


Figure 11: The auto naming window showing how to set the counters for a requirement

For more information on setting the auto-counters see the Set Auto Name Counters Help topic.

Where Auto-Naming is the preferred option the numbering in the name can be re-set using the package context menu option: right-click on a Package | Apply Auto-Naming to elements.

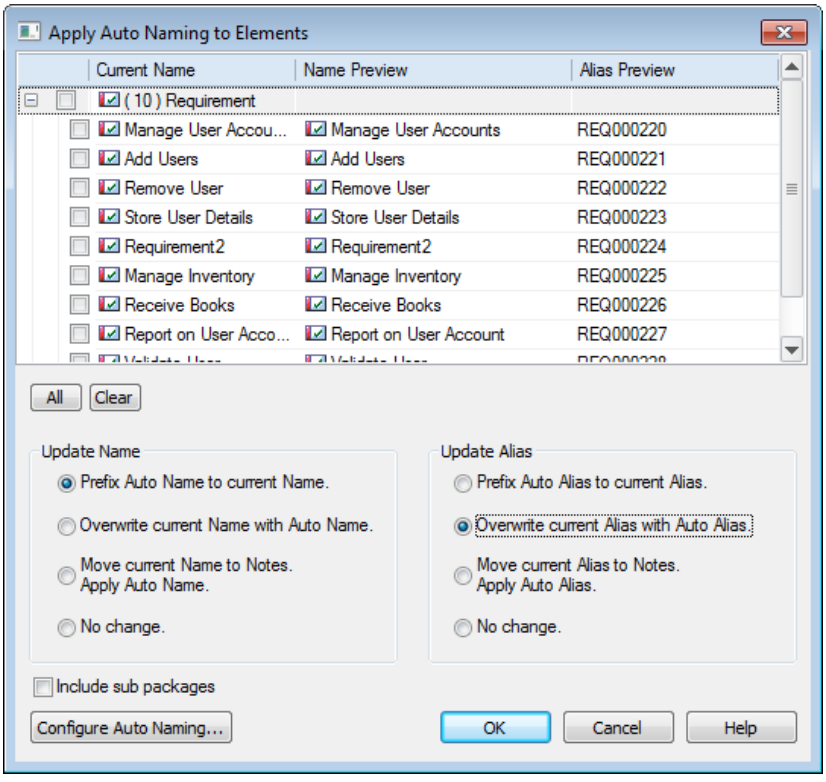


Figure 12: The dialog with the Auto-naming options

This feature can also be used for naming existing elements that are not yet Auto Named.

For information on using the auto-naming feature see the Auto-Naming Help topic.

Tips

To keep your requirement names separate from your requirement identifier it is best to use the **Alias** field for your Auto-Naming

Traceability and relating requirements

When modeling using requirement elements there are numerous UML connector types that can be used, however there are two types of relationship that are commonly used with requirement management. One for setting relationships between peer requirements (Aggregation), and another for representing how they will be implemented (for example a Realization by a Use Case).

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 and Classes. You can specify this relationship using the Realization link. A model element is marked as 'Realizing' a requirement. Once this link exists, Enterprise Architect 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).

Creating and Viewing Relationships

In Enterprise Architect, there are four key methods used for tracking requirements and forming relationships between the requirements and their related elements. These relationships define how those requirements are to be implemented within the system. The four 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 or the Quicklinker.
- 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 Traceability window provides a feature for tracing all the relationships of a selected element.
- Checking for unrealized requirements
  Using the Validation feature you can detect and view unrealized requirements.

Creating relationships using diagrams

Creating relationships between elements on a diagram is a simple process in Enterprise Architect. There are a number of methods you can use for this. Details on the most common methods are covered in detail in the Quick Linker and the Connect Requirements Help pages.

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 Browser, 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).

The Relationship Matrix

The Relationship Matrix allows you to create and view relationships, regardless of what diagram or package the elements are placed in. It can be used with any UML element, but it is particularly useful in Requirements Management for two reasons:

1) With a large system definition it may be cumbersome using diagrams to define large sets of relationships between requirements and other elements. An alternative is to use the Relationship Matrix to quickly set relationships without the need to draw these in a diagram.

2) As the development phase progresses, each element that defines either an Aggregation or Realization of a requirement, such as another requirement or a Use Case, must be linked to its root requirement definition using a connector. It is this linking that is critical to backward traceability. This is where the Relationship matrix can be useful tool for verification of links.

   Figure 13 is an example of two related requirements that are in separate packages.

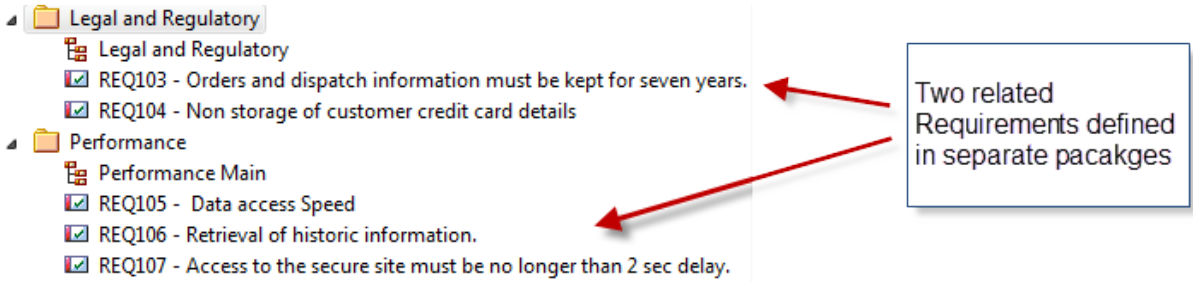


Figure 13: Requirements defined in separate packages

Figure 14 shows the Relationship Matrix view connection between the requirements in Figure 13. The source and targets are set up to show the 'Legal and Regulatory' package as the source and the 'Performance' package as the target.

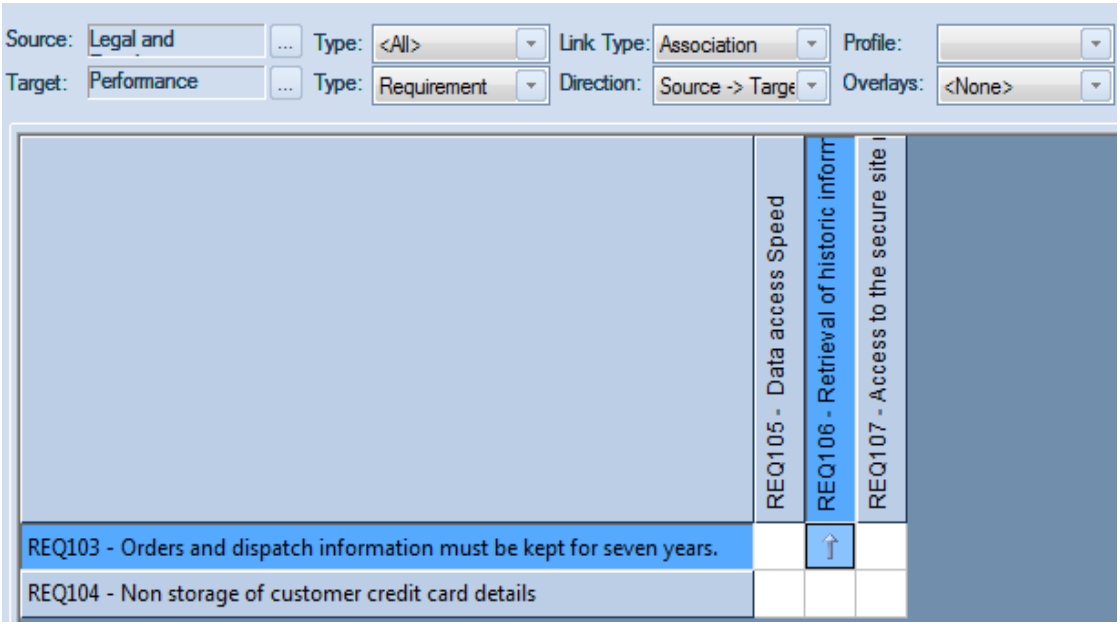


Figure 14: A Relationship Matrix view connections between Elements from different packages

For more details on adding a Relationship between requirements using the Relationship Matrix see the Creating and Deleting Relationships Help topic.

Tips and tricks

- Use the Relationships Matrix to create, edit and delete relationships, rather than doing this graphically in the model diagrams. This is most applicable when crossing different levels of abstraction e.g. from requirements to Use Cases.
- Save your favorite or commonly-used matrix profiles. These will then be listed in the Resources view. This is very useful because it may often be necessary to look at the same kind of relationships a number of times, and you can use the same settings without having to re-enter them.
- Use the automatic process of creating a relationship using drag-and-drop – see *Drag and Drop Realizations* section.
- After creating a relationship you can right-click on a requirement in a diagram and select Insert Related Elements. This will open a dialog to select any related elements to be placed in the diagram for you.

Using the Traceability window

The Traceability window allows you to view the relationships across a hierarchy of elements. It 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'.

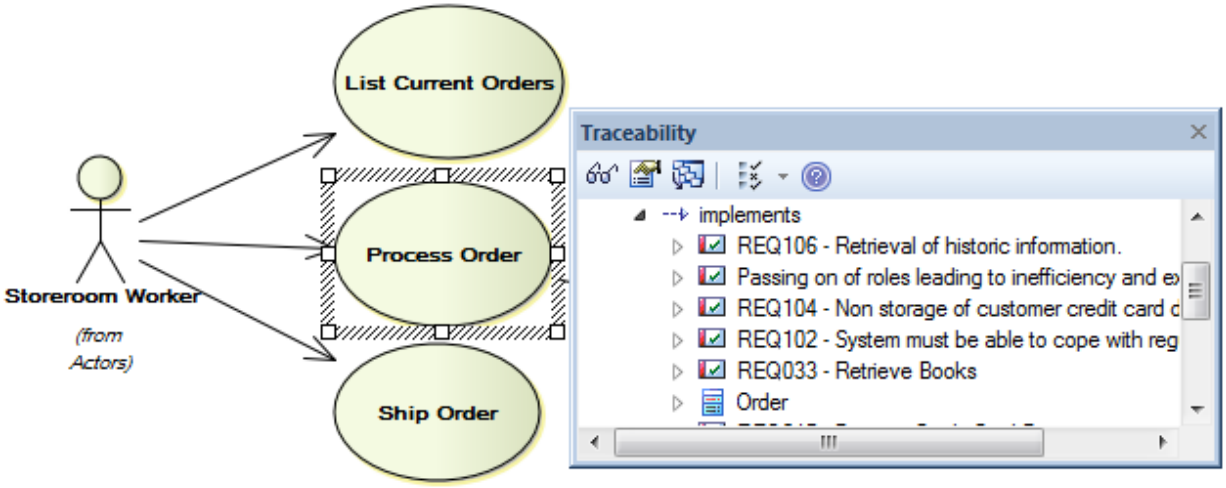


Figure 15: The Traceability window showing this Use Case's requirement relationships

To use the Traceability window for viewing relationships

- Open the Traceability window (View | Traceability or Ctrl+Shift+4).
- Select the element for which you want to display relationships.

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.

Checking for unrealized requirements

A useful option when dealing with large numbers of requirements it is the ability to check if any requirements have not been realized (for example one not yet realized by a Use Case). The Model Validation option supports checking for unrealized elements as show in Figure 16.

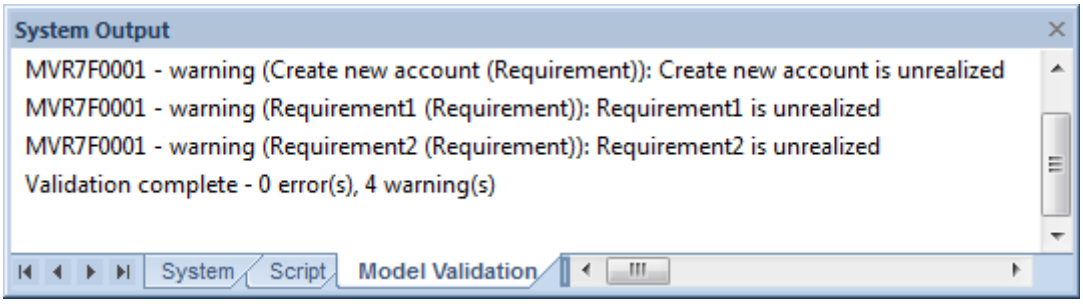


Figure 16: Using Model Validation to check for unrealized requirements

To access the Model Validation feature see the Model Validation Help topic.

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, and the prior state of the model. This can be particularly useful for recording a history of changes to requirements models.

Figure 17 is an example of viewing alterations to an element directly in the Audit View. This shows a number of alterations with the first selected to show the details on the right pane.

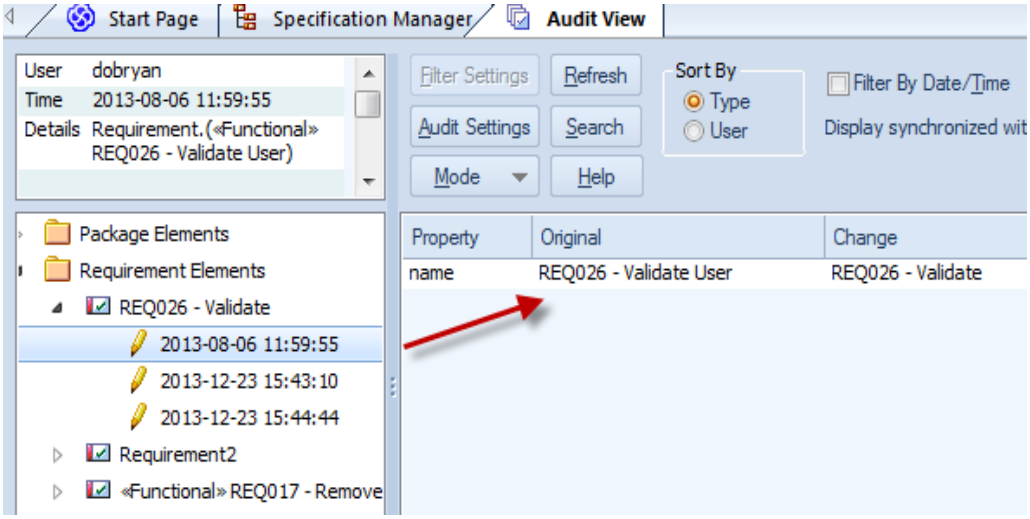


Figure 17: Audit view showing a list of alterations with the details of a Name change shown

With the Auditing View enabled the System Output | Audit History window can be used to show the list of changes for the selected element. Figure 18 shows a requirement selected in the Specification Manager and a set of alterations to this element logged in the Audit History view.

The System Output view can be accessed from the main menu View | System Output **(Ctrl+Shift+8).**

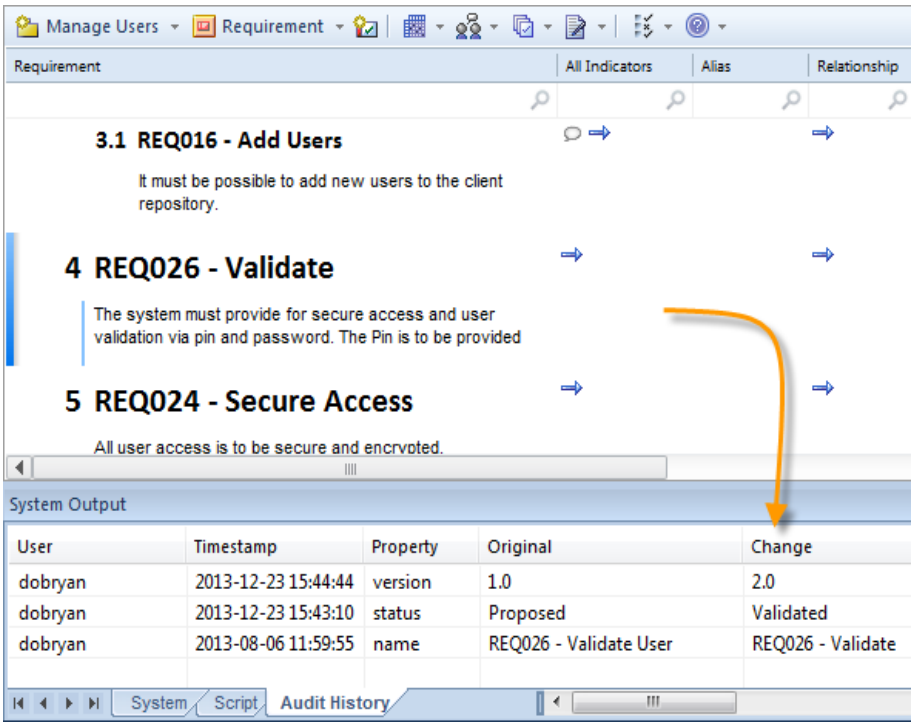


Figure 18: Audit History view open to highlight a change in the Specification Manager

For more information on using the Auditing features see the Auditing Help topic.

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 (such as by month, phase, version or build). Baselines can then be compared to the current model and changes selectively rolled back.

Baselines can also be used for 'Branching' by creating a duplicate repository (a Branch). After updating the requirements model in the Branch repository the changes can be merged back to the source repository using the 'Load other Baselines' feature.

For more information on setting up baselines and viewing differences see the Package Baselines Help topic.

Change Requests and Issues on External Requirements

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

a) Using the **Maintenance** window to list Changes, Defects, Issues and Tasks against each element.

b) 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 window

The **Maintenance** window 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** window can be accessed from the main menu using: Element | Maintenance or (Alt+4). Figure 19 is an example of a set of changes listed for an element:

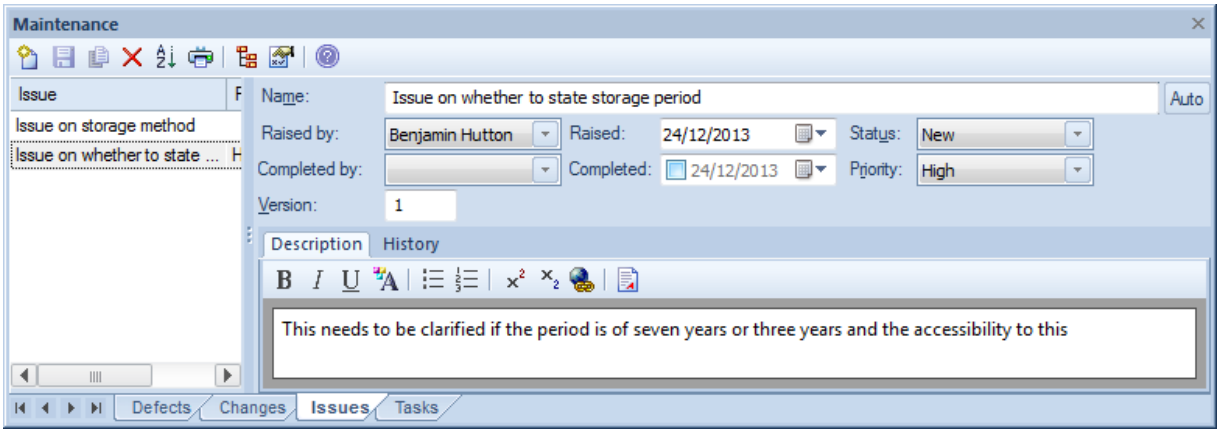


Figure 19: Maintenance view showing Issues lodged against a Requirement.

The common use of the **Maintenance** window in Requirements Management is for logging - internal to the requirement element – any detailed Requirement-Issues and Change-Requests. These can also be logged by linking to external elements of these same types.

Using Maintenance elements for Changes and Issues

Enterprise Architect's maintenance elements include elements of type: Issue and Change. These are accessible from: Toolbox | More Tools | Custom.

Maintenance elements can be linked using a connector to any element 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.

Figure 20 illustrates the use of an Issue element associated with a requirement.

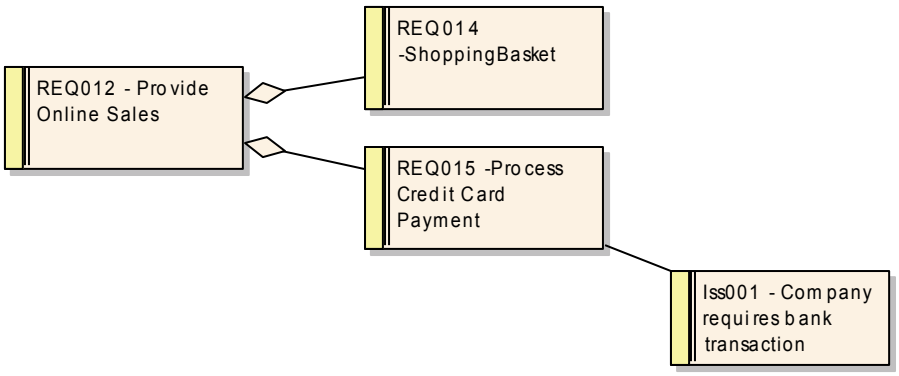


Figure 20: Using Issue Elements

Internal Requirements

As an alternative to using requirement elements, Enterprise Architect allows you to enter requirements within 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). Figure 21 displays a single requirement defined within a Use Case element.

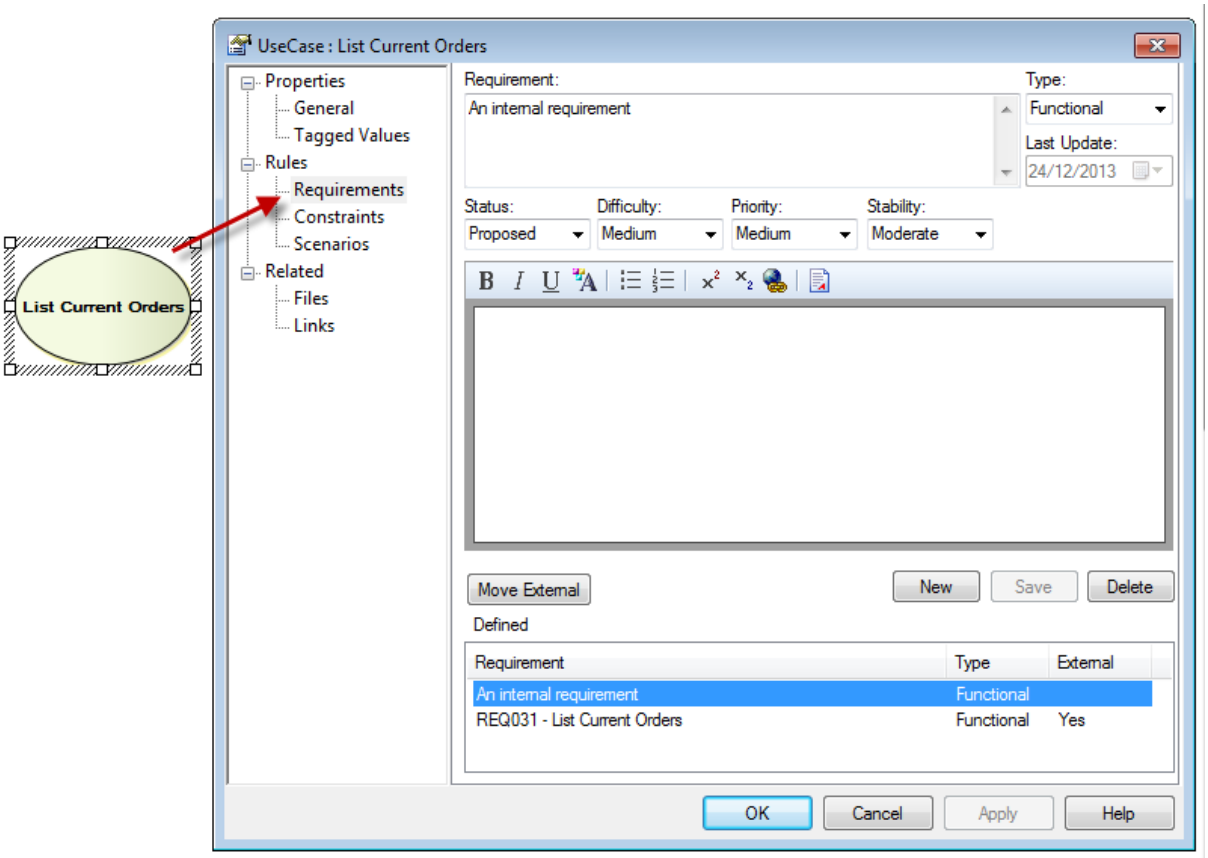


Figure 21: An example of Internal Requirement lodged on a Use Case

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 elements, such as Use Cases, gives a simple introduction into the more complex requirements definitions using external requirements.

This feature became a trend very early in UML modeling. Although the use of external requirements on a higher level of abstraction to use cases has become more popular, the internal requirements can still be a useful feature.

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'.
- 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 quality requirements documentation

A definition of a requirement 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 definition can be generated.

External Requirements reports

Enterprise Architect's Document Report Generator includes a report template for external requirements. This can be easily copied and modified to suit your reporting needs. Figure 22 shows the details of a standard requirements report.

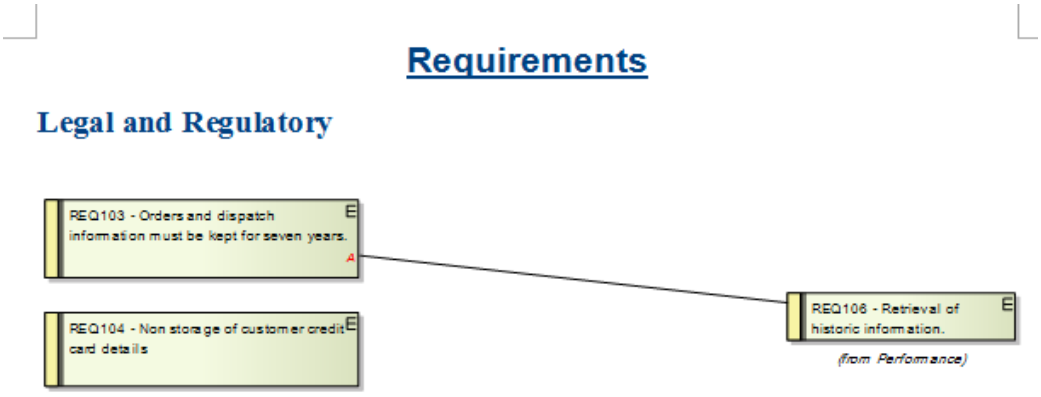


Figure: Legal and Regulatory

REQ103 - Orders and dispatch information must be kept for seven years.			
<<Functional>>	*Status:* Proposed	*Priority:* Medium	*Difficulty:* High
	Phase: 1.0	*Version: 1.0*	
	There is a legal requirement that customer order and dispatch information must be kept for seven years. There should be a facility to archive these orders and ensure that they are not deleted before seven years from the creation date.		

REQ104 - Non storage of customer credit card details			
<<Functional>>	*Status:* Proposed	*Priority:* Medium	*Difficulty:* Medium
	Phase: 1.0	*Version: 1.0*	
	There is a legal requirement that customer credit card details including the credit card number must not be stored online.		

Figure 22: An example of a default requirements report

To copy an existing document template for editing see the **New** option under **Manage your Custom templates** on the RTF Templates Help page.

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.

Implementation report, Dependency report, and the Package Browser

There are two additional reports, as well as the Package Browser view, that are very useful when managing requirements.

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 | QA Reports & Metrics and click on the Implementation Details tab.

Common uses

- To locate all elements that should have realizations.

- To locate all elements that 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 the system 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 | QA Reports & Metrics and click on the Dependency Details tab.

Package Browser view

The Package Browser view can be used to get a quick, simple and clear picture of the requirements and their detailed text. The Package Browser view shows the textual description of the elements in the Package tree.

To view the Package Browser, select from the main menu: View | Package Browser.

Ensure the View notes option is set using the View Notes icon: 

Using the context menu there are also options to create RTF reports or directly print text reports from the Package Browser.

Additional Requirements Management features

Enterprise Architect provides a number of other features for Requirements Management, as explained below.

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: Settings | Project Types | General Types and click on the Requirement tab as displayed in Figure 23.

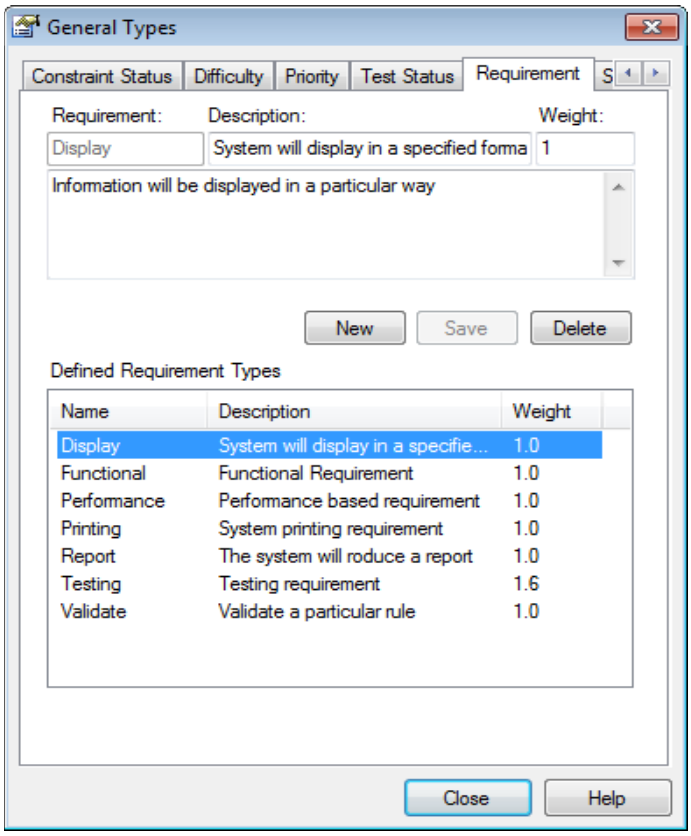


Figure 23: Configuring user-defined Requirement types

- 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*.

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 see the Color Code External Requirements Help topic.

Common Uses

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

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. Enterprise Architect 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

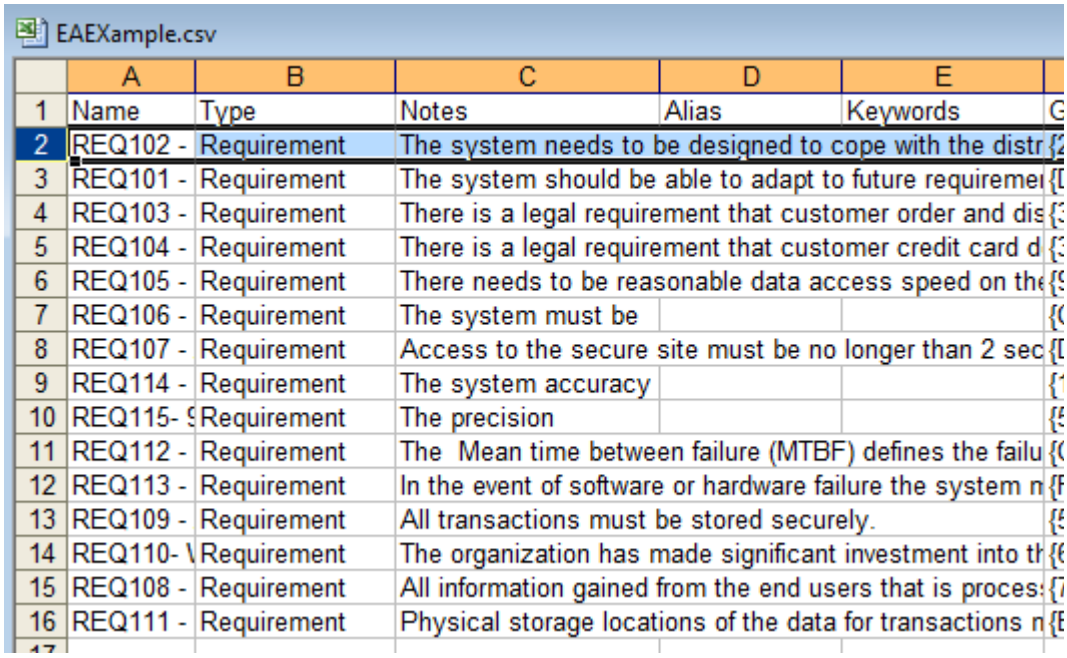
Where you need to import requirements from an external source there are a number of features that can be used for importing, including:

1. CSV import (from a spreadsheet)
2. Creating Requirement elements by dragging text from a document
3. Importing a document to an internal Linked Document and creating new elements hyperlinked to the text in the document

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.

Figure 24 is a simple example of a spreadsheet containing a set of requirements to be imported into Enterprise Architect.


EAEXample.csv

	A	B	C	D	E	
1	Name	Type	Notes	Alias	Keywords	G
2	REQ102 -	Requirement	The system needs to be designed to cope with the distr			{2
3	REQ101 -	Requirement	The system should be able to adapt to future requireme			{[
4	REQ103 -	Requirement	There is a legal requirement that customer order and dis			{3
5	REQ104 -	Requirement	There is a legal requirement that customer credit card d			{3
6	REQ105 -	Requirement	There needs to be reasonable data access speed on the			{9
7	REQ106 -	Requirement	The system must be			{(
8	REQ107 -	Requirement	Access to the secure site must be no longer than 2 sec			{[
9	REQ114 -	Requirement	The system accuracy			{1
10	REQ115-	Requirement	The precision			{5
11	REQ112 -	Requirement	The Mean time between failure (MTBF) defines the failu			{(
12	REQ113 -	Requirement	In the event of software or hardware failure the system n			{F
13	REQ109 -	Requirement	All transactions must be stored securely.			{5
14	REQ110-	Requirement	The organization has made significant investment into th			{6
15	REQ108 -	Requirement	All information gained from the end users that is proces			{7
16	REQ111 -	Requirement	Physical storage locations of the data for transactions n			{E

Figure 24: An example requirements spreadsheet

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 Enterprise Architect

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 | Model Import/Export | CSV Import/Export Specifications.

This will return the following window:

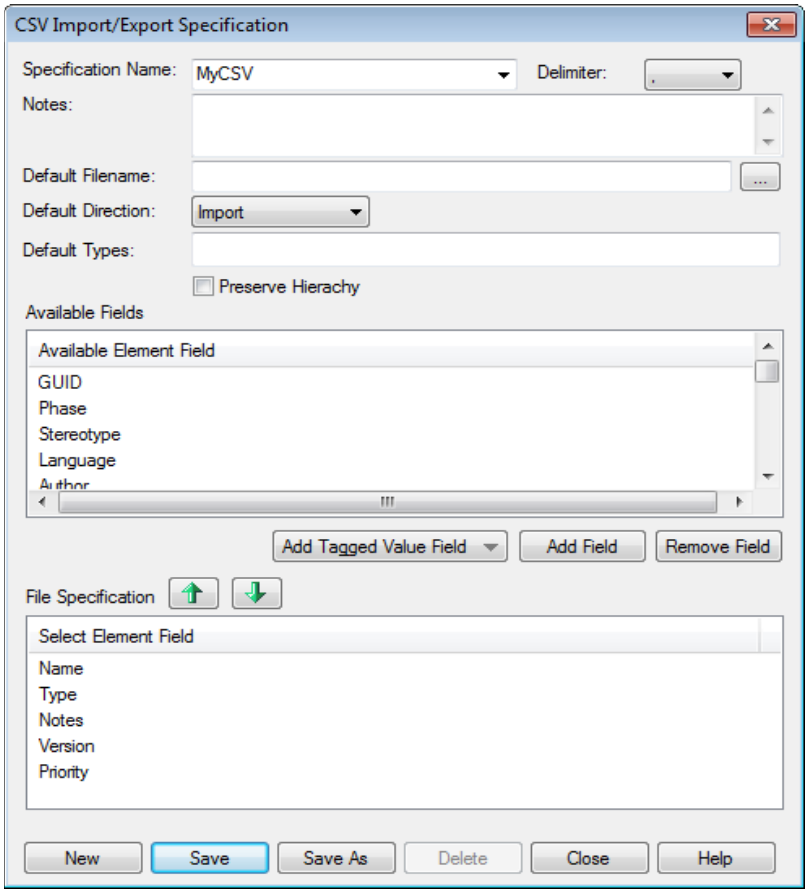


Figure 25: The CSV import specification

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 the Add Field 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 | Model 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 fields as shown in Figure 26:

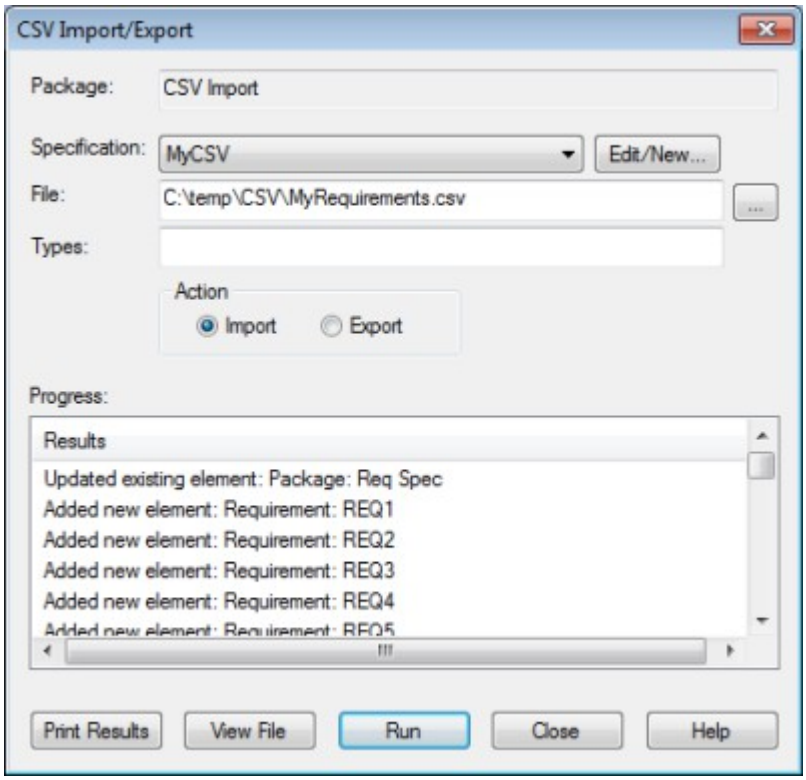


Figure 26: an example of the CSV import process

- **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. Figure 27 is a Project Browser view of the data imported (via CSV) from the spreadsheet above.

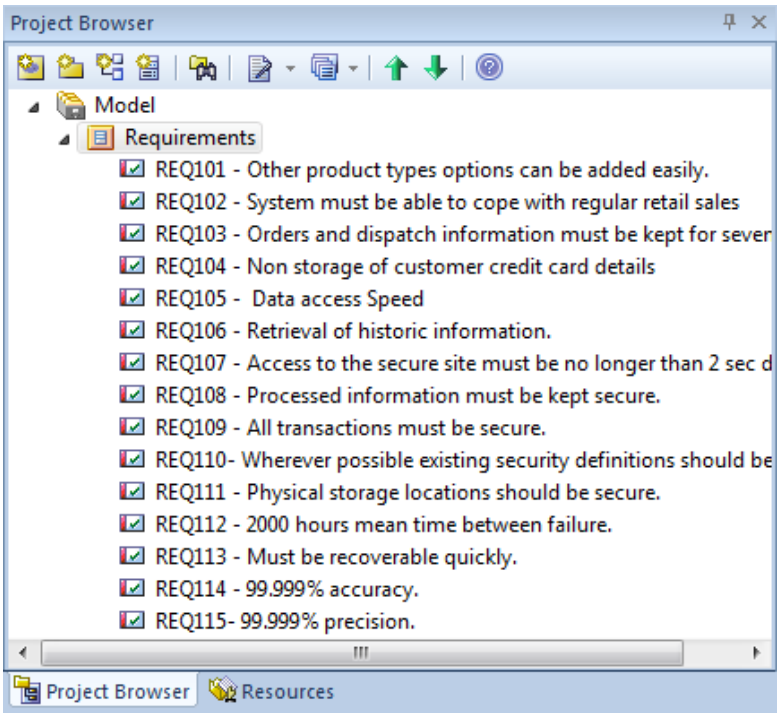


Figure 27: Requirements imported via the CSV import

Once imported, the requirements can then be placed in different packages by dragging the elements in the Project Browser 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 the: [x] Preserve hierarchy option as shown in Figure 28:

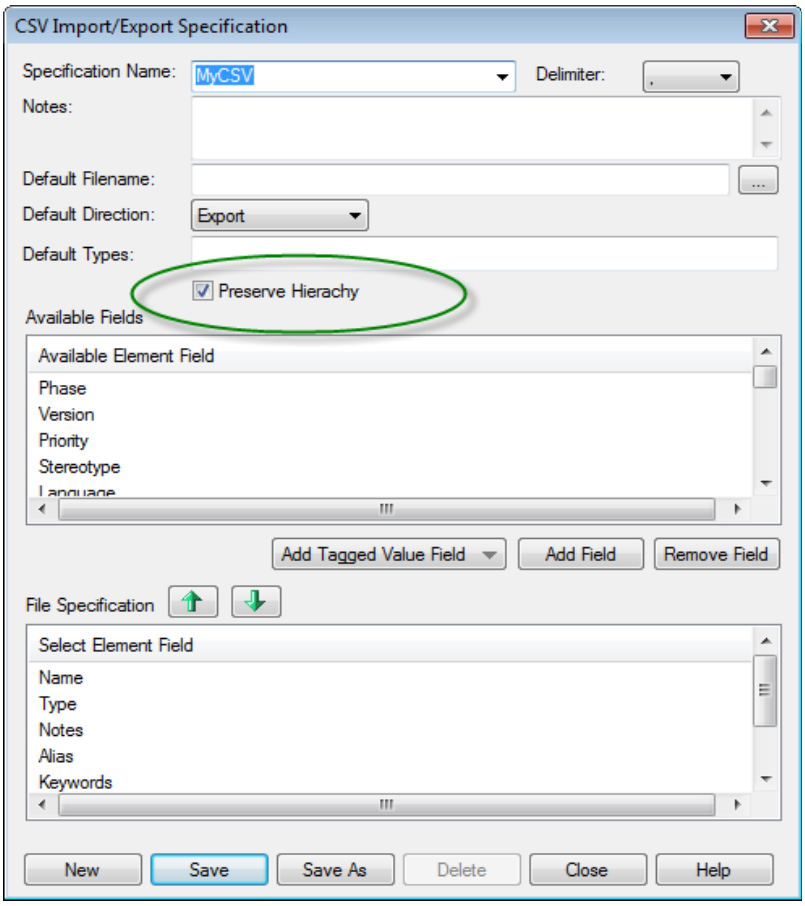


Figure 28: CSV specification with Preserve Hierarchy set

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
Req Spec	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

Figure 29 is a Project Browser view of the hierarchy imported using the above CSV file.

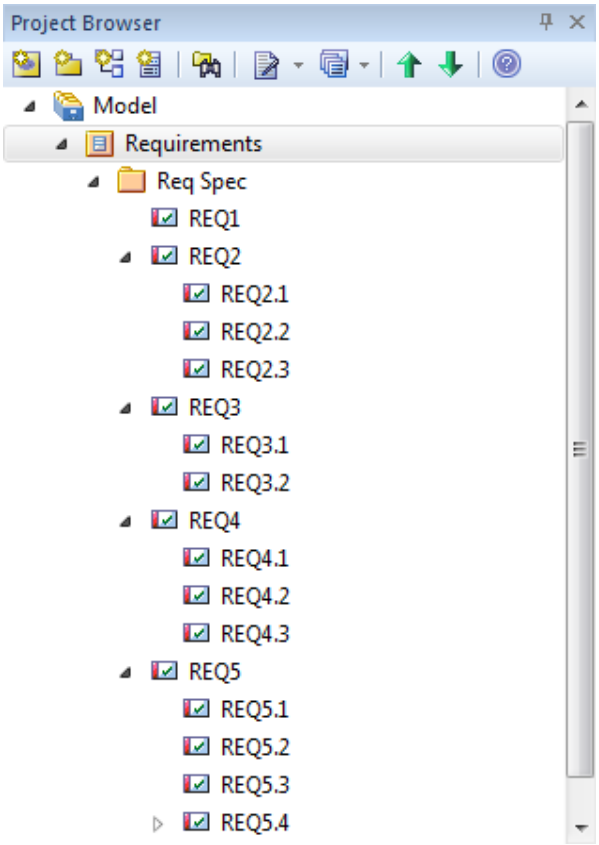


Figure 29: Project Browser view of imported CSV text

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.

Where there needs to be a more automated means of importing requirements in CSV format, supplied with Enterprise Architect are base-level scripts that provide a foundation for your own user-defined method of importing in a CSV format. For more details see the Scripting view (Tools | Scripting) EAScriptLib | Jscript-CSV

Note: ensure the EAScriptLibrary is available under Settings | MDG Technologies.

Figure 30 shows the Scripting view containing the CSV script.

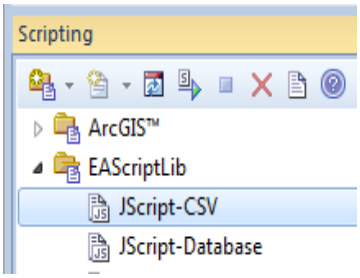


Figure 30: Scripts available for automating your own CSV import/export

Dragging text from a document

In the early stages of analyses, requirements may be defined in a text document. Enterprise Architect allows you to drag text from a document to create a Requirement element.

To do this you simply block the header and text in the external application and drag this onto a Requirement diagram to create a Requirement element. The first line of text is passed to the element name. Any other lines of text are passed to the notes.

Figure 31 shows an example of an element created from a text document (green), along with the steps for dragging text from an external document onto a diagram to create new requirement elements (orange line).

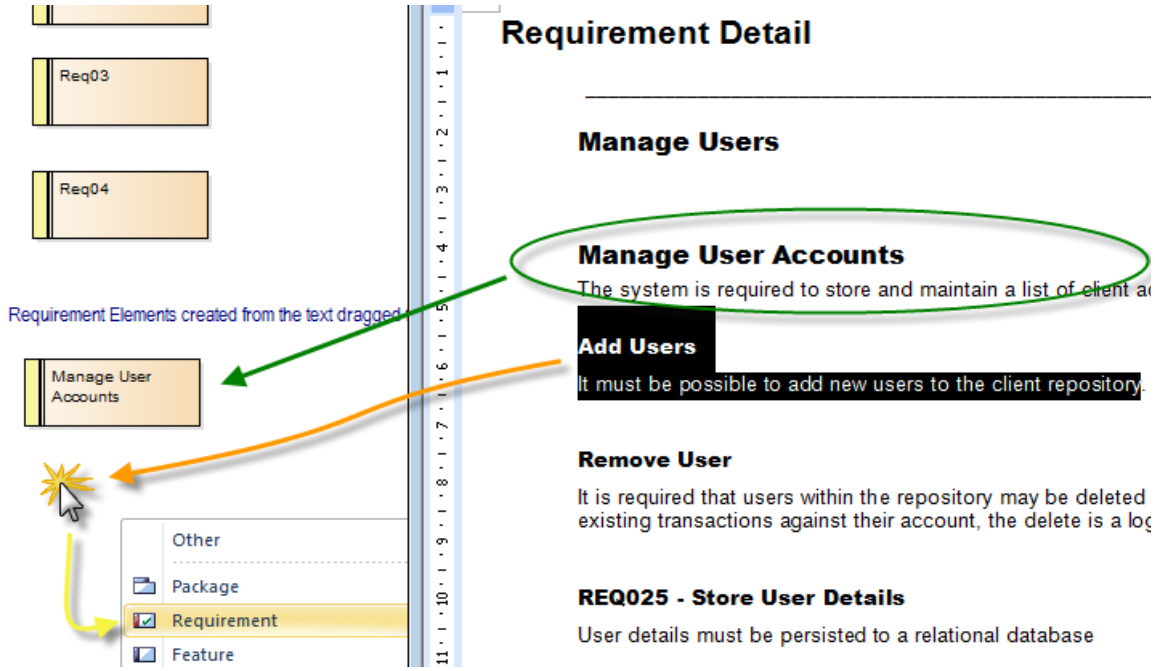


Figure 31: Creating elements by dragging text from an external application

Tips and tricks

This is very useful for selectively importing text from a document with requirements grouped amongst other detail not required in the model.

Creating hyperlinked elements from a Linked Document

For a third option you can import the document as a UML 'Document Artifact' or as the 'Linked Document' of an element. This can be performed by simply dragging a file onto a diagram.

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 that are subsequently defined in the semantic model (such as detailed notes, constraints and status). Figure 32 is an example of the source text in a Linked Document and the elements created from, and linked to the text.

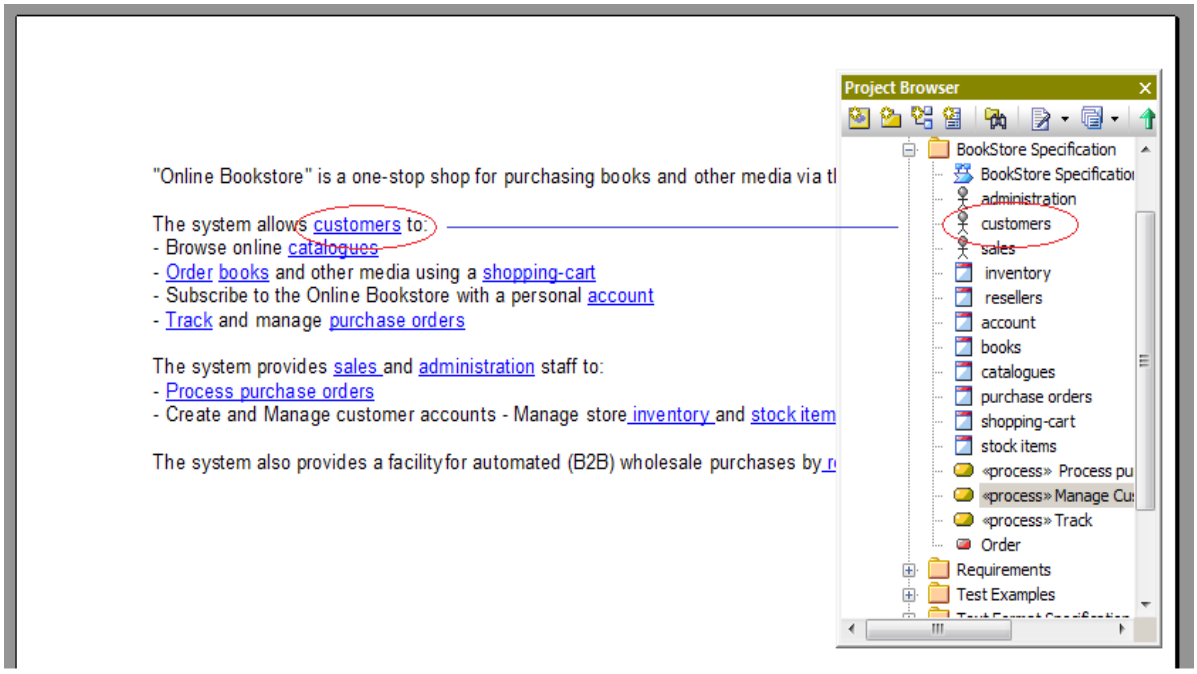


Figure 32: Creating Elements from unformatted text in a Linked Document.

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

Attaching documents and files

A UML based specification, although outwardly graphical, allows for textual descriptions for each element. If you have purely text based documents that need to be referenced, these can be linked to the element (a requirement, a use case, etc.), using a number of different options:

1) Elements and the Files tab.
   External files can be linked to an element using the Files tab in the element properties window. See the Associated Files Help topic.

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.

4) Dragging a file onto a Diagram. See the '**Create File Artifact**' Help topic.

   When dragging a file from say the Windows File Explorer on to a diagram you will be given the option to create an Artifact element as either an internal storage or external link to the file (option 1 above). The internal option stores the file as an OLE object that can be opened by double-clicking on the Element created.

> **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 Linked Document editor menu. This import option is accessible by using a right-click on the body of the document and selecting: File | Import.

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.

An introduction to Use Cases in Enterprise Architect

Use Cases are used to model single tasks a user of the system might 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 features described 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. Each diagram 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.

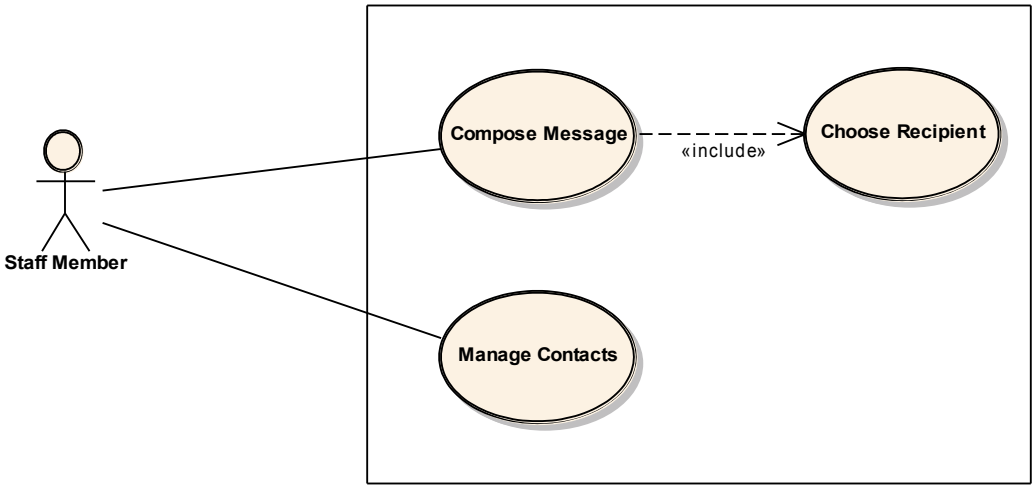


Figure 33: Use case diagram showing actors and use cases.

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 on a package in the Project Browser and select Add | Add new Model From Wizard > Basic UML Technology | Use Case. This will provide you with a basic use case.

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 pages in 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 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.

Linking with requirements

Using the realize relationship, you can define which use cases are implementing the requirements. See the section Traceability and Relating Requirements for more information on creating these links.

Defining Scenarios

The use case Properties dialog allows you to specify Attributes that apply to the use case as well as detailing the scenarios. Double clicking on a use case element and selecting the Rule | Scenarios page allows you to define a structured specification of the scenarios covering a use case.

Figure 34 shows an example of the Basic Path for a Use Case Scenario.

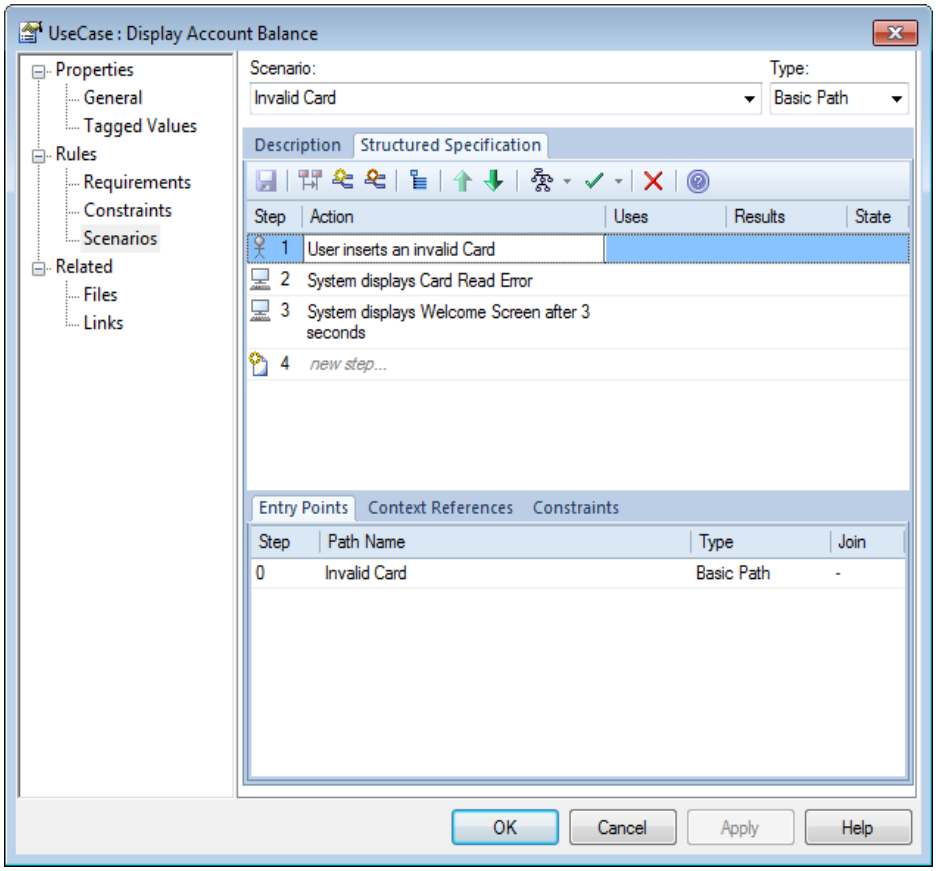


Figure 34: The Use Case scenario page

For more details on using the Structured Scenarios see the Scenarios Help page.

Additional features of Enterprise Architect

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

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.

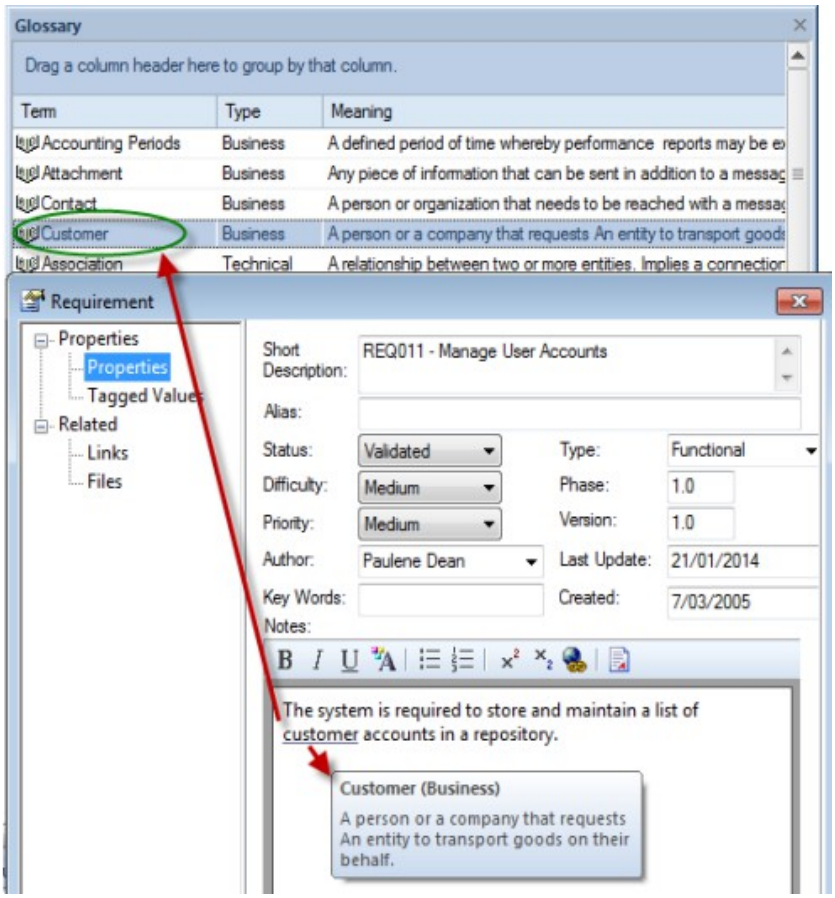


Figure 35: Element notes showing a reference to a glossary term and the glossary entry of this term.

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 Project | Model Import Export | Export Reference Data and Project | Model Import Export | Import Reference data).

- When creating new reports using the RTF generator consider adding the glossary as an appendix to some of the key documents to be generated.

For more details see the Project Glossary and the Notes Help pages.

Defining requirement Attributes using a Profile

As stated above, the requirement element can be predefined to include a set of user-defined Attributes. These are used to document user specific qualities. The additional Attributes can be defined using a *Profile Definition*.

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. When creating a Profile you can use model based Tagged Values or define these in the Profile meta-model.

To set up a pre-defined Tagged Value, select from the main menu: Settings | UML Types | Tagged Value Types. This will bring up the Tagged Values definition screentab as shown below.

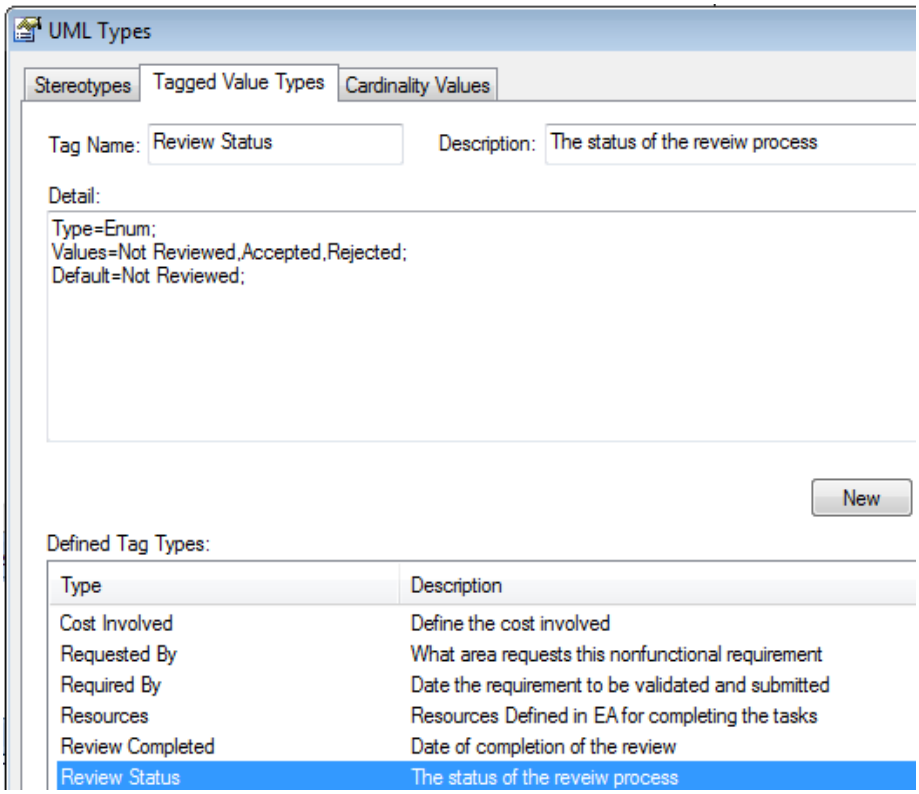


Figure 36: A Tagged Value definition

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;

Figure 37 shows this as viewed in the Tagged Values window as a drop-down option box.

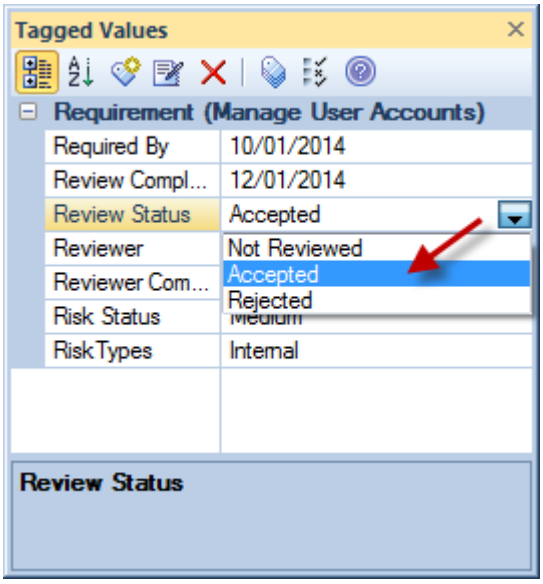


Figure 37: An Element's Tagged Values

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

For more information on setting up the standard types, and a list of types available see the Predefined Structured Types Help page.

Defining a Profile

Profiles allow you to define a set of extensions to standard UML elements using your own predefined Tagged Values. Using a Profile you can define multiple requirement types each with its own set of Tagged Values.

To define a new element type we use a Profile, created using the Profile Helpers. For more detail on using these see the Using Profile Helpers Help topic.

Figure 38 shows a simple Profile for creating an element type that includes Tagged Values. Two of these are drop-down selections (**Priority** and **ReviewStatus**).

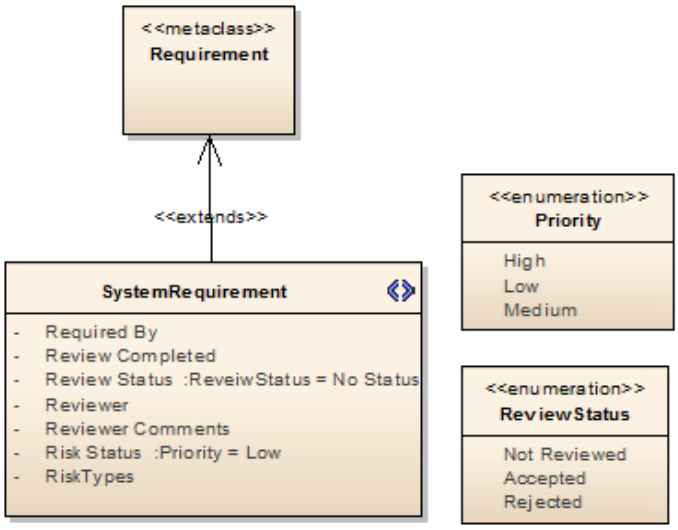


Figure 38: A Profile definition for a Requirement

To define the drop-down selections this profile includes two Enumeration elements (**Priority** and **ReviewStatus**). These are referred to within the Element.Attributes of **SystemRequirement**.

These Attributes are rendered as Tagged Values, as shown in Figure 39.

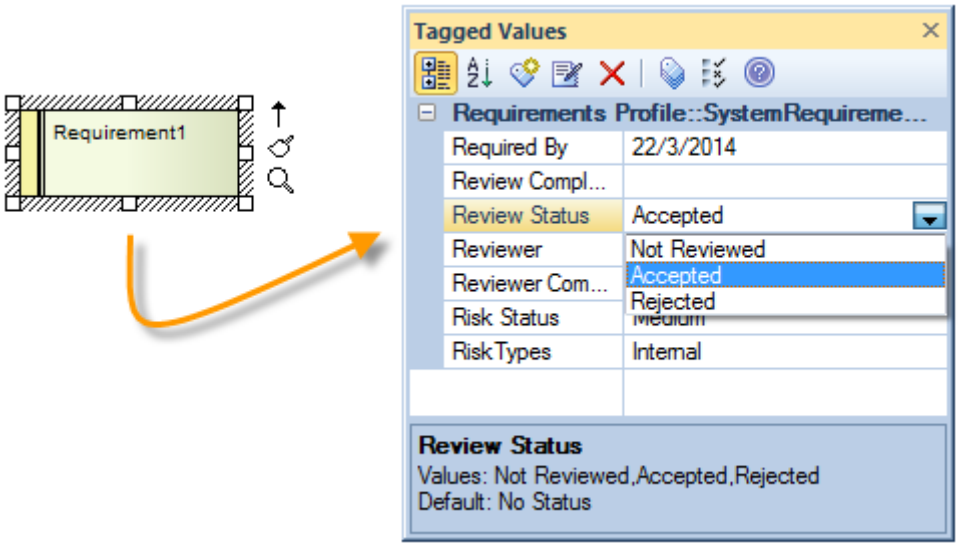


Figure 39: A SystemRequirement (see Figure 38), showing the Tagged Values

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.

For details on implementing this as an MDG technology across multiple repositories see the MDG Technologies SDK Help topic.

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 center of the Enterprise Architect interface.
- Attributes – Data fields containing information within requirement elements.
