IBM® Rational® Rhapsody® Gateway Add On

Rhapsody Coupling Notes

Rhapsody®

IBM® Rational® Rhapsody® Gateway Add On

Rhapsody Coupling Notes

License Agreement

The information in this publication is subject to change without notice, and Dassault Systèmes and its affiliates assume no responsibility for any errors which may appear herein. No warranties, either expressed or implied, are made regarding Rhapsody software and its fitness for any particular purpose.

Trademarks

Contents

Contents **5**

Preface **7**

Rhapsody Analysis **9**

- Importing Requirements 9
- Specific Parameters of a Rhapsody Based Type 11
- Setting Rhapsody Options 11
  - Display Labels or Elements Names 12
  - Select Multi-Project Behavior 12
- Capturing Rhapsody Tags in Rhapsody Gateway 13
  - Capture Generic Attributes 13
  - Capture Specific Attributes 13
  - Capture Enumerations 14
  - Capture Multi-valued Attributes 14

High Level Requirements Addition **15**

- Adding Requirements to Rhapsody 15
- Dealing with High Level Attributes 21
- High Level Requirements Synchronization Process 22
  - New Requirements 22
  - Modified Requirements 22
  - Deleted Requirements 23
- Reaching a Requirement 23

Requirements Traceability **25**

- Using Dependencies 25
- Using Rhapsody Anchors 26
- Links Created from Rhapsody Gateway 26

Exporting to DOORS **29**

- Creating Rhapsody Elements into DOORS 29
- Managing, Capturing and Exporting Tags 32
  - Exporting Generic Tags 32
  - Exporting Specific Tags 33
- Traceability and Links 33

Exporting to RequisitePro **37**

Support of Profiles **39**

- Rhapsody Gateway and SysML 39
- Rhapsody Gateway and DoDAF 41

Considering Other Profiles 42

Rhapsody CM Capabilities 45

Troubleshootings 49

Preserving Add High Level Requirements Logs 49

Preface

The Rhapsody Gateway is a full requirements traceability solution, integrated in the Rhapsody environment.

The project to be analyzed must be configured in Rhapsody Gateway, specifying the nature of the files to be processed (Word, FrameMaker, UML analysis tools, Design tools, etc.), their location, the links between them and the requirements traceability information to be searched for in each of these files.

Rhapsody Gateway analyzes the documents and produces an image of requirements traceability information in relation to the project as a whole. It is then possible to set reference versions that can be transmitted if necessary to a configuration management tool and to generate analysis documentation in customized formats.

Some standard operations on a Rational Rhapsody model such as import, navigation or export ("add high level"), required Rational Rhapsody (Rhapsody) to be installed.

Rhapsody Gateway provides the following features:

- a configuration editor for the definition of the project graph
- converters for importing different kinds of files to be analyzed (DOORS Modules, Word, Excel, FrameMaker files, UML tool files, etc.)
- an analysis interface in coverage mode, impact analysis mode and graphic view mode
- a filter editor for processing analysis results
- a version editor to detect and save requirements changes

These features are detailed in the *User Guide*.

In addition, Rhapsody Gateway provides for advanced users:

- a type editor for defining project traceability elements
- a template editor for generated documents

These features are developed in the *Customization Guide*.

This following document gives basic explanations that will help Rhapsody customers to work efficiently with Rhapsody Gateway.

Rhapsody Analysis

This chapter describes how Rhapsody Gateway captures information items from Rhapsody, and how these information are analyzed to be used as requirements, attributes, text, and so on.

High level requirements can be captured from various sources of documents such as Word, Excel, DOORS, RequisitePro, etc.

See the *Customization Guide* and the *Coupling Notes* to see specific details on the management of requirements capture from these sources.

When Rhapsody Gateway is launched for the first time from Rhapsody, a project is automatically created with the same name as the Rhapsody model. In this project, the Rhapsody model is already inserted.

This section describes how to import requirements to create a traceability project and some alternatives to configure the project.

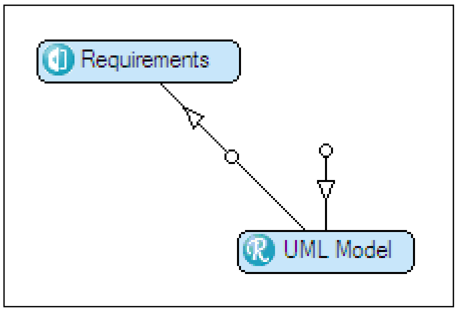
Importing Requirements

To insert requirements into this project's documents, a document containing high level requirements must afterwards be inserted.

To enclose this high level requirements document such as a DOORS module, follow these steps. (See **Adding a document** and **Adding Coverage Links between Documents** sections from the *User Guide* for more details).

1. Click **Add a document** to insert a new document into the Traceability Description Area.
2. From the form at the bottom of the Project editor window, select the DOORS type of analysis and the desired module to be analyzed.
3. Click **Add a cover** to add a coverage link from the Rhapsody model to the DOORS module.

The Traceability Description Area contains a project like the following:

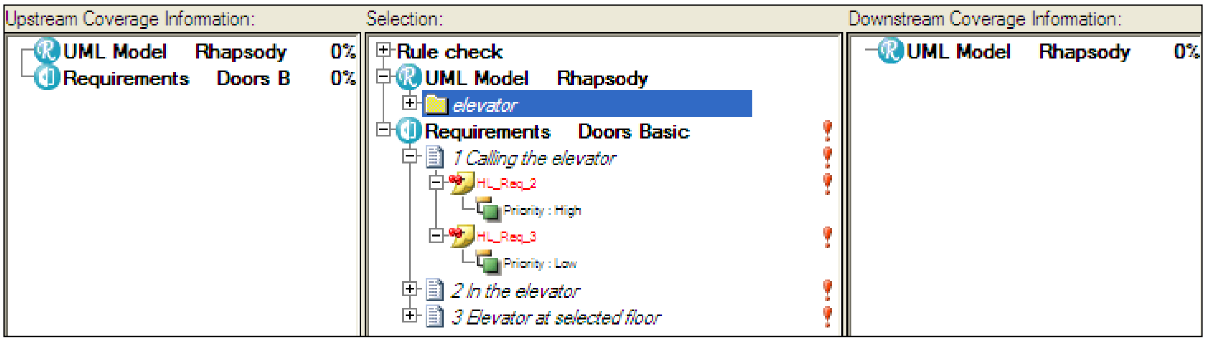


Validating the changes allows you to consider the requirements of these documents. Rhapsody Gateway can capture:

- requirement IDs
- requirement texts
- attributes
- sections

Sections are captured to keep the hierarchy of the high level document if the corresponding option is checked in the **Add high level requirements** option. See the **Requirements Creation into Rhapsody** section of this document to learn more. This is done as a result of creation of packages.

The analyzed information appears in the Rhapsody Gateway main window as follows.



Upstream Coverage Information:

UML Model Rhapsody 0%
Requirements Doors B 0%

Selection:

Rule check
UML Model Rhapsody
elevator
Requirements Doors Basic
1 Calling the elevator
HL_Req_2
Priority : High
HL_Req_3
Priority : Low
2 In the elevator
3 Elevator at selected floor

Downstream Coverage Information:

UML Model Rhapsody 0%

Note

The display of the captured requirements can be customized using the **Display text** field selected from the Types for the requirement. This customization is also used for the requirements display in Rhapsody, as explained later in this document.

Specific Parameters of a Rhapsody Based Type

When inserting a Rhapsody model into the Rhapsody Gateway project editor, you are able to specify additional parameters through variables available in the **Variable** list. When the Rhapsody document is selected in the project editor the following variables are available:

- **Language**–You can specify the used analysis language. Available values are C, C++, Ada and Java languages. By default, the language is C++.
- **Capture diagrams**–You can capture vectorized diagrams from Rhapsody elements such as sequence diagrams. For example, you can export these diagrams into DOORS.

Note

If there are a high number of diagrams to be loaded for a model, this may impact the performances.

- **Package**–You can reduce analysis from all the models to only one package and its sub-packages. Type the name of the package in the **Value** box.

Note

If the Rhapsody model contains an important number of packages and only one specific package is of interest for traceability, it is recommended to work only on the concerned package. This may improve the performances.

- **With stereotypes applied**–You can complete the XML intermediate file with stereotypes applied on each model element. This information is interesting when using an XML customized type.
- **With Rich Text**–You can get the rich text properties of a captured text, refer to the *Traceability Elements* > *Text* section of the *User Guide* documentation to know which rich text properties are supported. The text is displayed formatted in the Text area.

Setting Rhapsody Options

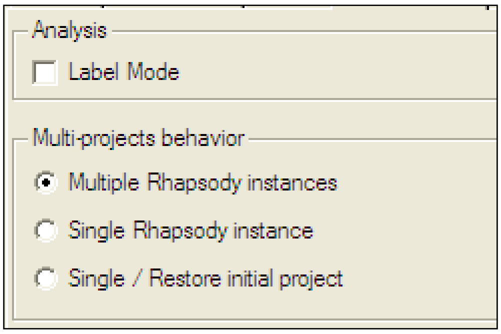
You can access some Rhapsody options from Rhapsody Gateway.

1. Select **Tools > Options**.

   The **Configuration** window is displayed.

2. Select the **Rhapsody** tab.

   The following options appear.



Analysis

□ Label Mode

Multi-projects behavior

● Multiple Rhapsody instances

○ Single Rhapsody instance

○ Single / Restore initial project

Display Labels or Elements Names

You can display either labels or element names, depending on a model browser setting (Label Mode).

1. Select **Label Mode** to display labels defined in the model.

   When the **Label Mode** option is not selected, element names are displayed by default.

2. Click **OK** and reload the model (right-click UML Model and select the **Reload** option) to take into account the changes, so the analysis results provide the requested information.

Note (Advanced)

You can freeze the **Label Mode** for some particular Rhapsody Model. This setting, called “withLabels”, is currently not accessible in the GUI but can be added to the Project Editor.

To configure this setting, add the following variable/value to the proper section of the .rqtf file:

```
[UML Model]
…
Variable#Name=withLabels
Variable#Value=1
```

Once the .rqtf file is modified, reload the project (**File** > **Reload All**).

Select Multi-Project Behavior

You can provide GUI for the Rhapsody connection mode setting.

- **Multiple Rhapsody instances**—Select this option to create a temporary Rhapsody instance to convert unloaded Rhapsody projects. This option is selected by default unless another setting is specified.
- **Single Rhapsody instance**—Select this option to load every project in the current instance of Rhapsody.

- **Single / Restore initial project**—Select this option to load every project in the current instance of Rhapsody and to have the original project restored at the end of the operation.

Capturing Rhapsody Tags in Rhapsody Gateway

In Rhapsody, **Tags** are used to add information to the model base.

In Rhapsody Gateway, there are two types of attributes that can also be used in enumerations:

- **Boolean attributes** (i.e. Critical), they are created with a blank value in the Rhapsody model.
- **Valued attributes** (i.e. Priority = High), they are created with a value completed with the attribute value (i.e. “High”) in the Rhapsody model.

Capture Generic Attributes

A **generic attribute** is an attribute, which captures both an attribute name **label** and a value **identifier**. Other attributes are named **specific**.

The types provided by Rhapsody Gateway capture most Rhapsody tags using the generic attribute type named **Tag**. In most cases, it is necessary to define other attribute types.

The generic tag type can only capture String values, for other configurations you need to define a Specific attribute type.

Capture Specific Attributes

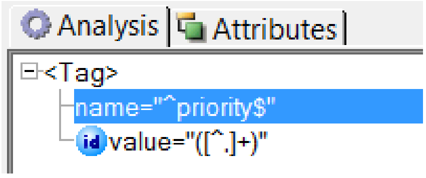
In some cases, you may have to define a specific attribute type with a particular configuration especially when exporting to other tools.

To create a specific attribute, customize your type with an appropriate attribute and define a condition for the tags to be caught. See *Creating a Customized Type of Analysis* section of the *Customization Guide* to have detailed information.

1. In your customized type, right-click `Tag` attribute then select **Duplicate**.

   A new tag appears.

2. Rename this tag with the same name as your tag in Rhapsody, i.e. `priority`.
3. In the **Analysis** tab, enter `^priority$` for the name and create a value identifier with for example `([^.]+)` for the value.



```
<Tag>
  name="^priority$"
  value="([^.]+)"
```

Tip:You can enter `$t: $i` in the **Display** box.

With the definition of attributes presented in the following example, only tags named "priority" are captured in the model.

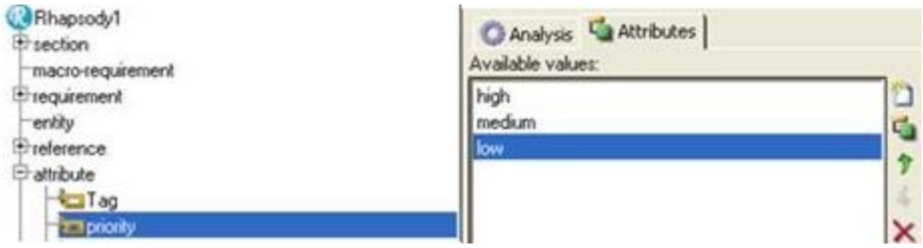
Important

To avoid the attributes to be captured twice when you are defining a specific tag, exclude the specific attributes in the generic Tag attribute: in the line `name=”^(?!BodyStart).”` add the specific tag name such as follows: `name=”^(?!BodyStart|Priority).”`

Capture Enumerations

If you are creating an enumerated attribute, you have to define all the enumeration values in the type.

From the **Attributes** tab, click **Add a new enumeration value** to create new values.



In the Coverage Analysis View, you can see multi-tagged values shown as separate attributes.

Capture Multi-valued Attributes

The import of multi-values attributes is available for enumeration and non-enumerated types.

From the **Attributes** tab, select **Is multi-valued** at the bottom of the configuration pane.

High Level Requirements Addition

The traceability requirements coming from the third level document need to be re-entered in Rhapsody to be covered by model elements.

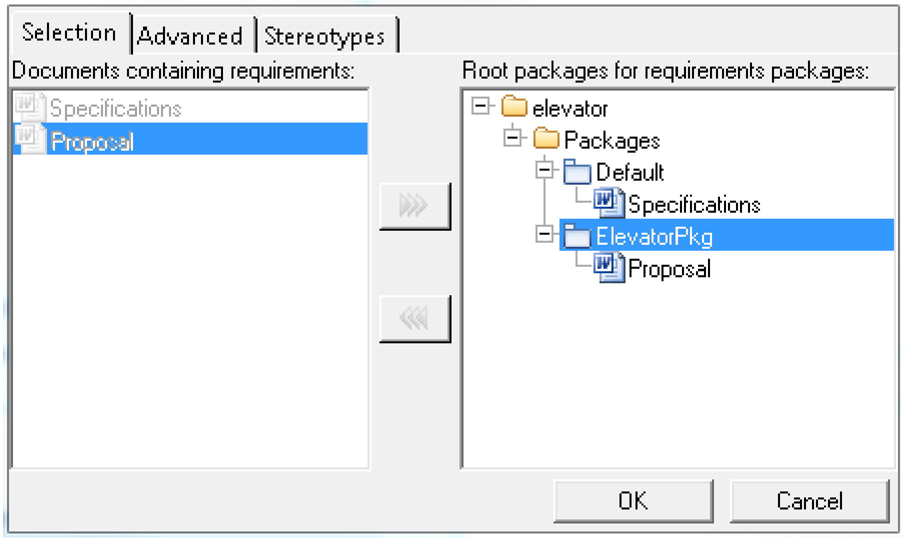
This section develops the corresponding process.

Adding Requirements to Rhapsody

You can perform the requirements addition once requirements have been captured from the high level document.

1. Select the root of a Rhapsody-based document, then select **Tools** > **Add high level requirements**.

The **Add high level requirements** dialog box appears:



2. In the **Selection** tab, you can select:

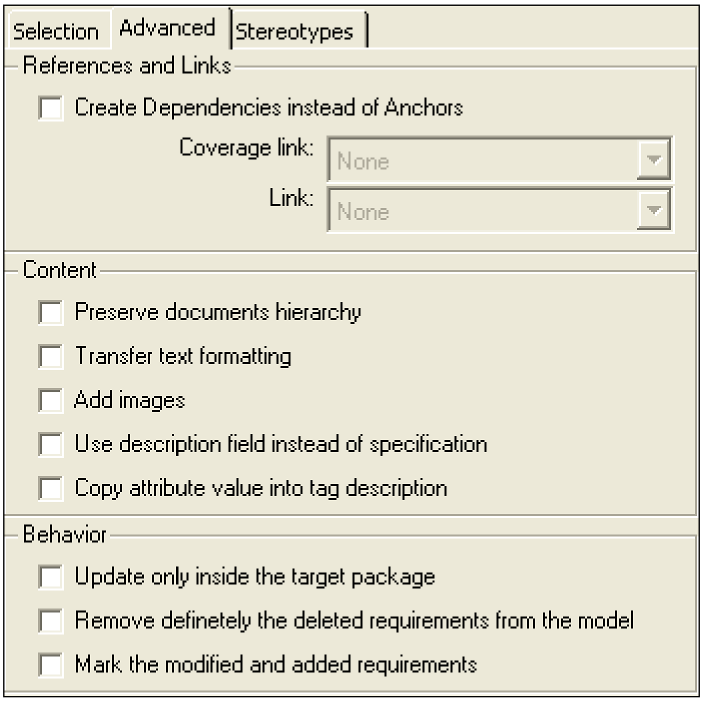
    - The requirements to import, if you have several documents covered by the Rhapsody model.
    - One or more packages for requirements in the Rhapsody model tree.

In the **Documents containing requirements**, select a document then select package in the tree of the **Root packages for requirements** area. Click » to associate the selected document with the selected package.

To dissociate a document from its associated package, select the document in the Root package for requirements area, then click «.

Tip: you can also drag and drop documents to create associations.

3. In the **Advanced** tab, you can specify how the links will be created in the Rhapsody model and how the document's hierarchy can be materialized into Rhapsody.



References and Links

☐ Create Dependencies instead of Anchors

Coverage link: None

Link: None

Content

☐ Preserve documents hierarchy

☐ Transfer text formatting

☐ Add images

☐ Use description field instead of specification

☐ Copy attribute value into tag description

Behavior

☐ Update only inside the target package

☐ Remove definetely the deleted requirements from the model

☐ Mark the modified and added requirements

- **Create Dependencies instead of Anchors**—Specifies whether or not links should be created as dependencies or as anchors in Rhapsody.
  By default, this option is not selected.
  When **Create Dependencies instead of Anchors** is selected, select the stereotypes of the links created as dependencies in the additional list boxes: **Coverage Link** and **Link**.

- **Preserve documents hierarchy**—Specifies whether or not the hierarchy of the documents should be kept when exporting into Rhapsody. If you choose to preserve the hierarchy, some packages based on sections' names are added during the 'Add high level' operation.
  By default, this option is not selected.

- **Transfer text formatting**—Inserts text of high level requirements with rich text formatting. To capture rich text, the **With Rich Text** option must have been selected for the high level document when this option is available.

- **Add images**—Adds images in Rhapsody such as those that come from DOORS.
  By default, this option is not selected.

- **Use description field instead of specification**—Enables the display of text from Rhapsody Gateway into the **Description** tab instead of the

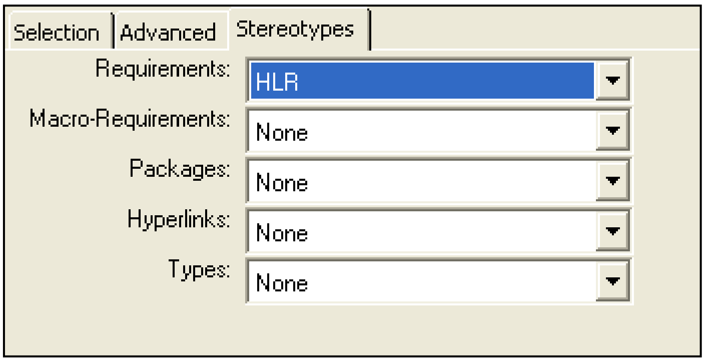
Specification field. The **Description** tab supports unicode characters (such as Chinese characters, Greek characters, etc.).
By default, this option is not selected.

- **Copy attribute value into tag description**—Enables the copy of the tag value of high level requirements into the tag description field.
- **Update only inside the target package**—Enables you to control the scope of the requirements synchronization. Rhapsody Gateway performs a global synchronization when this option is not selected. Thus, the **Add high level** requirements updates or adds the deleted marks on requirements that have moved outside of the target package.

  Tips: This option may be selected when the synchronization must be performed locally. The local synchronization behavior might be useful when synchronizing the documents one by one.

- **Remove definitely the deleted requirements from the model** option— If some requirements are deleted at high level, automatically remove requirements from the model instead of marking them with "Deleted_At_High_Level" tag.
- **Mark the modified and added requirement** option—Select this option to tag new or modified requirements with Added_At_High_Level or Modified_At_High_Level.

4. In the **Stereotypes** tab, you can specify the stereotypes for added high level requirements, macro-requirements, packages, hyperlinks and types. Values are suggested according to the project content. For compatibility reasons, the stereotype “fromXXX” where XXX is the add high level document type, is still added at the same time.

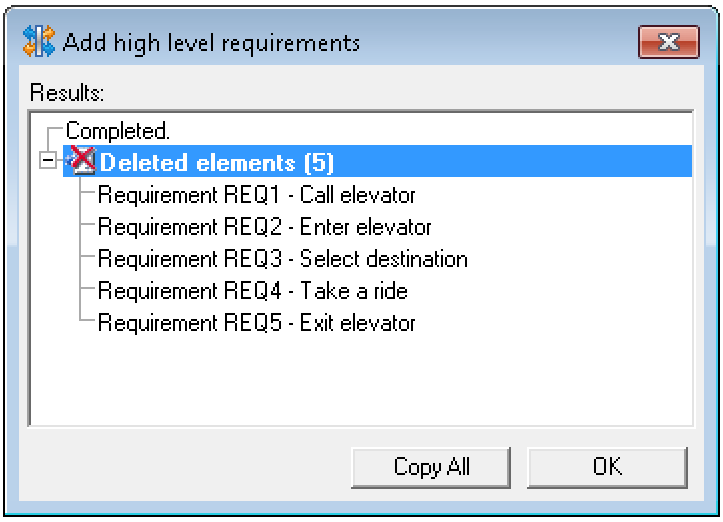


Requirements: HLR
Macro-Requirements: None
Packages: None
Hyperlinks: None
Types: None

Important

User actions will not be checked by Rhapsody Gateway **nor by Rhapsody**.
Rhapsody allows you to create several requirements with the same name.
If you move the requirements package manually in the Rhapsody tree, you can duplicate requirements in Rhapsody.
If you select a new location for the export, you can move imported requirements into another package depending on your answer to the move message.

5. At the end of the adding operation, a report message is displayed with the number of added, modified and marked as deleted requirements.



Add high level requirements

Results:

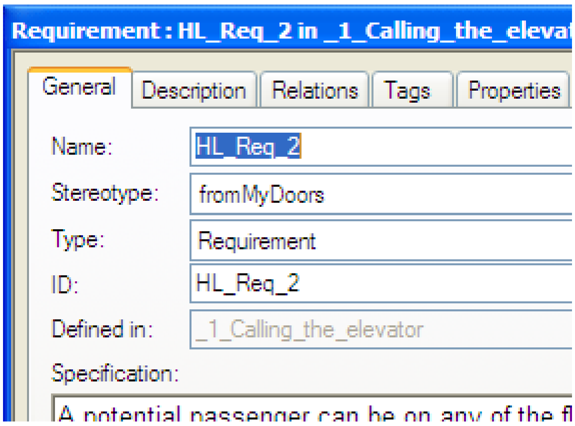
Completed.
Deleted elements (5)
Requirement REQ1 - Call elevator
Requirement REQ2 - Enter elevator
Requirement REQ3 - Select destination
Requirement REQ4 - Take a ride
Requirement REQ5 - Exit elevator

6. In the Rhapsody model, a new tree is created in the folder you have selected in the Add high level requirements dialog box. The imported requirements will be displayed in : `<selected folder>\Packages\<third level document name>\Requirements`.

Each requirement that comes from Rhapsody Gateway is created in Rhapsody with the following information:

- The Requirement ID goes to the Rhapsody ID field for requirement.
- The Requirement Text goes to the **Rhapsody Specification** field for requirement if the 'Use description field instead of specification' option is not checked otherwise the text is displayed into the Description pane.
- The Display (defined by the Display field of the requirement type) goes to the Rhapsody Name for requirement. If Display field is blank in the Rhapsody Gateway description type, both Name and ID fields in Rhapsody will contain the ID information.
- The Type of analysis of the requirements document is referenced in the Stereotype field. The value of the Stereotype is defined by the from<Type> string, where <Type> is the type applied to the requirements document in the Rhapsody Gateway Project Editor.

The following picture displays this information:



Requirement : HL_Req_2 in _1_Calling_the_elevat

Name: HL_Req_2

Stereotype: fromMyDoors

Type: Requirement

ID: HL_Req_2

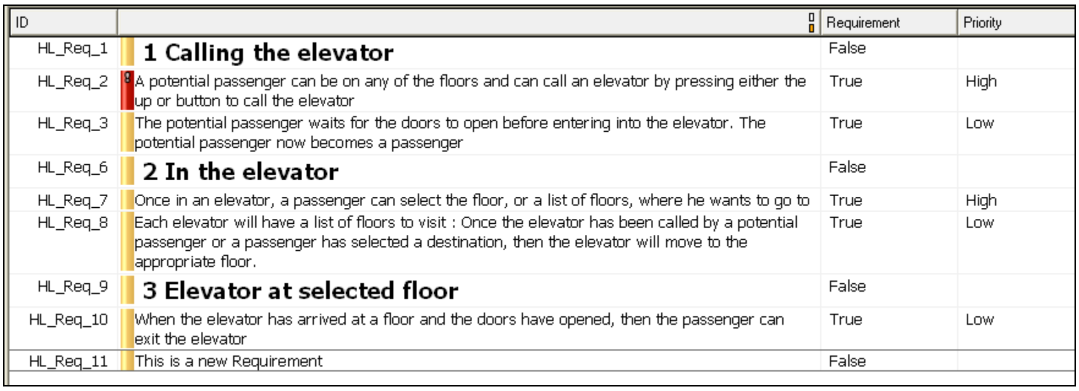
Defined in: _1_Calling_the_elevator

Specification:

A potential passenger can be on any of the fl

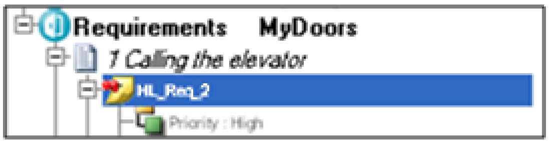
See below an example of Rhapsody requirements creation coming from DOORS:

View of the DOORS module

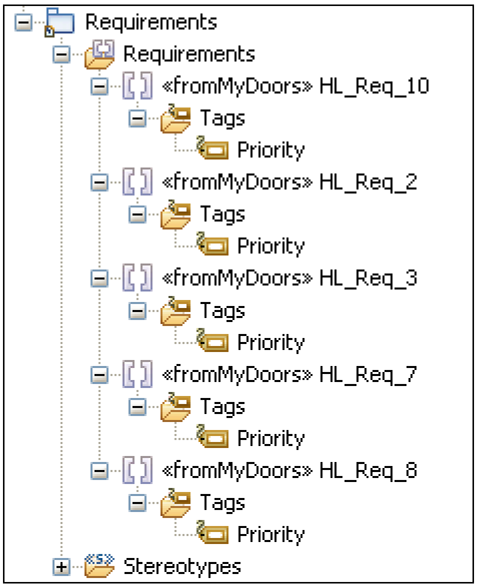


ID		Requirement	Priority
HL_Req_1	**1 Calling the elevator**	False	
HL_Req_2	A potential passenger can be on any of the floors and can call an elevator by pressing either the up or button to call the elevator	True	High
HL_Req_3	The potential passenger waits for the doors to open before entering into the elevator. The potential passenger now becomes a passenger	True	Low
HL_Req_6	**2 In the elevator**	False	
HL_Req_7	Once in an elevator, a passenger can select the floor, or a list of floors, where he wants to go to	True	High
HL_Req_8	Each elevator will have a list of floors to visit : Once the elevator has been called by a potential passenger or a passenger has selected a destination, then the elevator will move to the appropriate floor.	True	Low
HL_Req_9	**3 Elevator at selected floor**	False	
HL_Req_10	When the elevator has arrived at a floor and the doors have opened, then the passenger can exit the elevator	True	Low
HL_Req_11	This is a new Requirement	False	

Extract of the corresponding view in the Rhapsody Gateway tree



Corresponding view in the Rhapsody tree



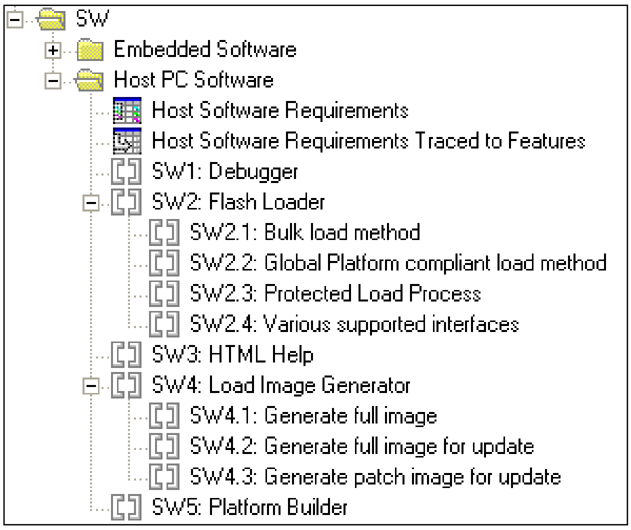
```
Requirements
    Requirements
        «fromMyDoors» HL_Req_10
            Tags
                Priority
        «fromMyDoors» HL_Req_2
            Tags
                Priority
        «fromMyDoors» HL_Req_3
            Tags
                Priority
        «fromMyDoors» HL_Req_7
            Tags
                Priority
        «fromMyDoors» HL_Req_8
            Tags
                Priority
    Stereotypes
```

Note

If the **Preserve documents hierarchy** option is selected, some packages are created to keep the hierarchy of the high level document in Rhapsody.

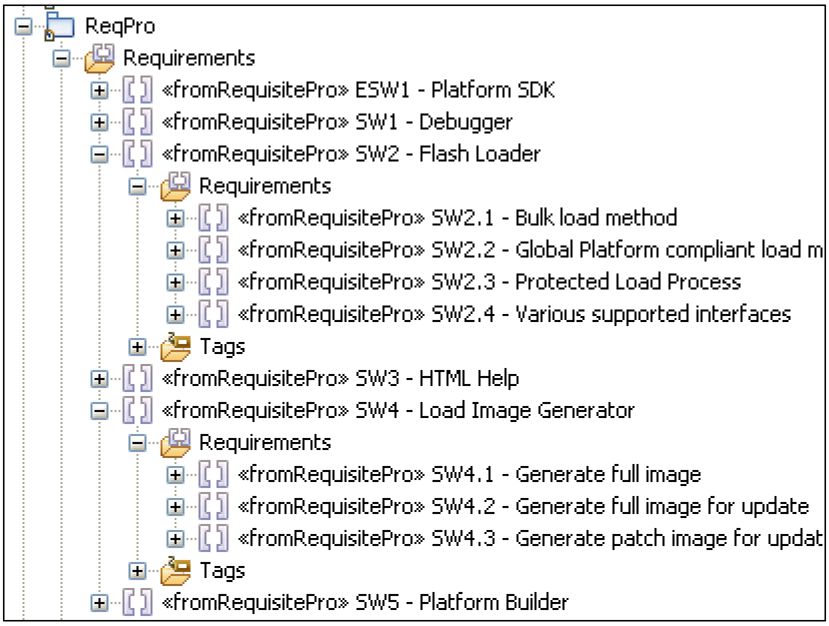
See below an example of Rhapsody requirements creation coming from RequisitePro:

View of the RequisitePro tree



```
SW
    Embedded Software
    Host PC Software
        Host Software Requirements
        Host Software Requirements Traced to Features
        SW1: Debugger
        SW2: Flash Loader
            SW2.1: Bulk load method
            SW2.2: Global Platform compliant load method
            SW2.3: Protected Load Process
            SW2.4: Various supported interfaces
        SW3: HTML Help
        SW4: Load Image Generator
            SW4.1: Generate full image
            SW4.2: Generate full image for update
            SW4.3: Generate patch image for update
        SW5: Platform Builder
```

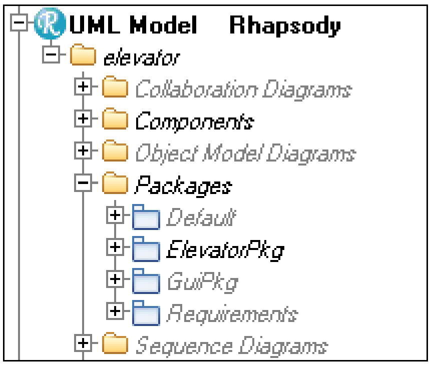
Corresponding View of the Rhapsody tree



- ReqPro
  - Requirements
    - «fromRequisitePro» ESW1 - Platform SDK
    - «fromRequisitePro» SW1 - Debugger
    - «fromRequisitePro» SW2 - Flash Loader
      - Requirements
        - «fromRequisitePro» SW2.1 - Bulk load method
        - «fromRequisitePro» SW2.2 - Global Platform compliant load m
        - «fromRequisitePro» SW2.3 - Protected Load Process
        - «fromRequisitePro» SW2.4 - Various supported interfaces
      - Tags
    - «fromRequisitePro» SW3 - HTML Help
    - «fromRequisitePro» SW4 - Load Image Generator
      - Requirements
        - «fromRequisitePro» SW4.1 - Generate full image
        - «fromRequisitePro» SW4.2 - Generate full image for update
        - «fromRequisitePro» SW4.3 - Generate patch image for updat
      - Tags
    - «fromRequisitePro» SW5 - Platform Builder

Important Note

The transfer of requirements from a high level document to Rhapsody is a great help for Rhapsody users. However, this operation does not represent a traceability action. Although Rhapsody Gateway manages requirements traceability, it does not display the high level requirements created by this operation. In a Rhapsody tree, the **Requirements** section is grayed. Requirements are presented only in the high level document.

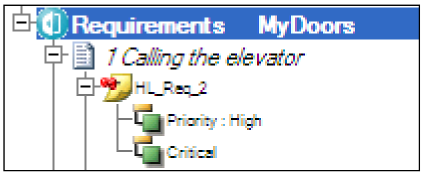


- UML Model Rhapsody
  - *elevator*
    - *Collaboration Diagrams*
    - *Components*
    - *Object Model Diagrams*
    - *Packages*
      - *Default*
      - *ElevatorPkg*
      - *GuiPkg*
      - *Requirements*
    - *Sequence Diagrams*

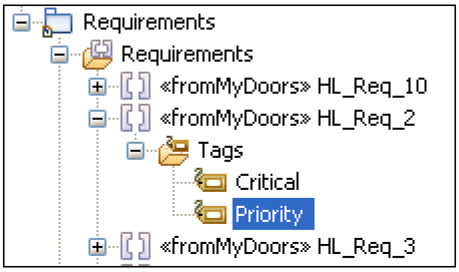
Dealing with High Level Attributes

Rhapsody Gateway attributes attached to requirements captured from high level documents are created as Rhapsody **Tags** according to their attribute types configuration.

In Rhapsody Gateway, the attributes are displayed as in the following screenshot.



The corresponding representation in Rhapsody looks like the following example.



Add high level requirements supports the addition of Boolean attribute types (with no value captured), String attribute types and Enumerated attribute types.

It also supports the addition of multi-valued String and Enumeration attribute types.

Tip: Make sure that attribute types are properly configured for high level document types otherwise tags may not be created as expected.

High Level Requirements Synchronization Process

The traceability process consists of successive phases. In fact, the UML model and the third party tool document would be likely to change at any time. So this is helpful to visualize the requirements which have been added or changed during the process.

This section summarizes the requirements status during the process activities.

New Requirements

Added requirements are created into Rhapsody using the **Add high level requirements** feature as explained in the previous section. This happens when you handle a new Rhapsody Gateway project or when you add new requirements in your high level documents.

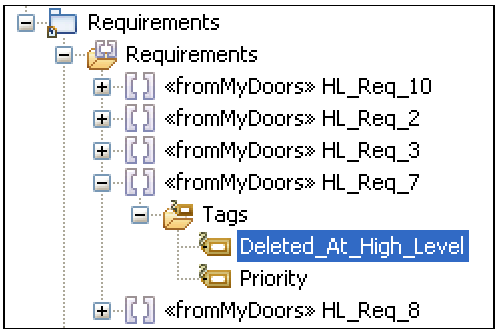
Modified Requirements

If requirements change, the modifications are automatically detected by Rhapsody Gateway, as described in details in the *User Guide*. As with every time you synchronize, Rhapsody notices these changes when you launch an update by using the **Add high level requirements** feature.

During this process, all the modified information are updated. For instance, the **Description** fields are updated and the requirement text is changed.

Deleted Requirements

If some requirements are deleted at a high level, Rhapsody Gateway does not directly delete the requirements in the Rhapsody model. Rhapsody Gateway only adds a **Deleted at high level** tag on this kind of requirements, as shown in the following picture.



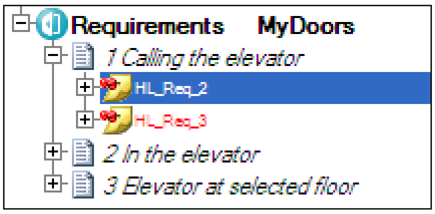
Requirements
Requirements
«fromMyDoors» HL_Req_10
«fromMyDoors» HL_Req_2
«fromMyDoors» HL_Req_3
«fromMyDoors» HL_Req_7
Tags
Deleted_At_High_Level
Priority
«fromMyDoors» HL_Req_8

It is up to the Rhapsody user to delete the model elements according to the appropriate process. An **Add high level** option can automate this deletion.

Reaching a Requirement

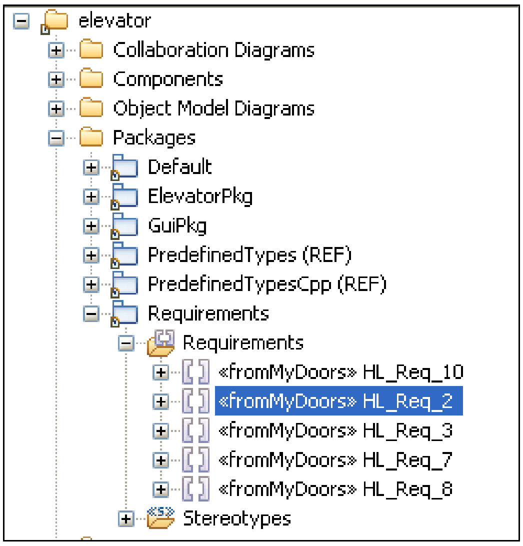
In Rhapsody Gateway, a specific navigation feature is available to reach and highlight a high level requirement in Rhapsody.

1. From Rhapsody Gateway, right-click a requirement then select **Find in Rhapsody**.



Requirements MyDoors
1 Calling the elevator
HL_Req_2
HL_Req_3
2 In the elevator
3 Elevator at selected floor

2. In Rhapsody, the selected requirement is highlighted, such as follows:



elevator
Collaboration Diagrams
Components
Object Model Diagrams
Packages
Default
ElevatorPkg
GuiPkg
PredefinedTypes (REF)
PredefinedTypesCpp (REF)
Requirements
Requirements
«fromMyDoors» HL_Req_10
«fromMyDoors» HL_Req_2
«fromMyDoors» HL_Req_3
«fromMyDoors» HL_Req_7
«fromMyDoors» HL_Req_8
Stereotypes

Requirements Traceability

Requirements need to be associated with model elements. Several UML elements such as **use cases**, **sequential diagram** or **methods** can provide coverage information.

Traceability requirements are classically performed in the Rhapsody environment using anchors or dependencies.

This section describes the process of creating requirements from Rhapsody and from Rhapsody Gateway.

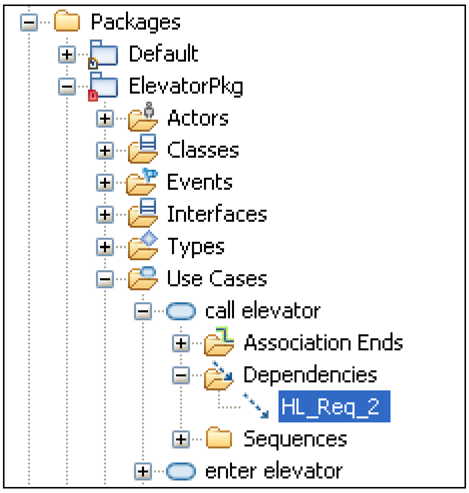
Using Dependencies

The default **Rhapsody** type of analysis captures as traceability information the Rhapsody **dependencies**. The stereotype of a dependency specifies the way in which the requirement relates to the model elements. See the **Advanced** pane of the **Add high level** box.

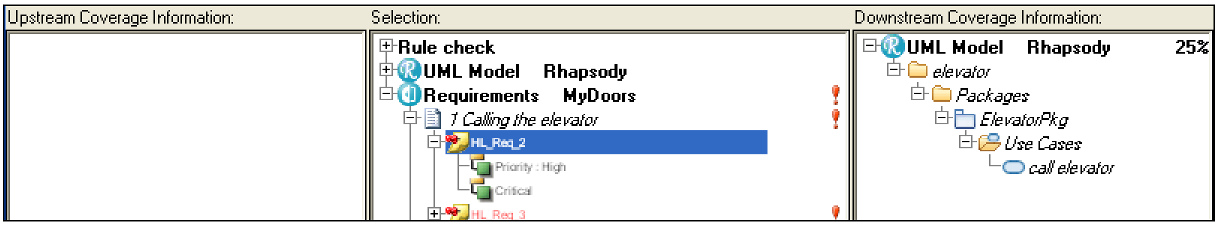
Dependencies with the following stereotypes: **trace**, **verify** and **satisfy** are captured as **References**, otherwise they are captured as **Links**. Refer to the Types editor to look at the default Rhapsody type properties.

To create coverage, follow these steps:

1. Choose a covering element. Create a dependency link with one of the previously listed stereotype to cover a requirement by the UML element.



2. Save the Rhapsody model, and then reload the Rhapsody Gateway project. The coverage is displayed in the Rhapsody Gateway main window.



Each time Rhapsody Gateway reloads the project, it presents the traceability between the Rhapsody model and the requirements.

See the section concerning “Support of SysML and DoDAF” in this document for detailed information. See the *Rhapsody User Guide* for more information.

Using Rhapsody Anchors

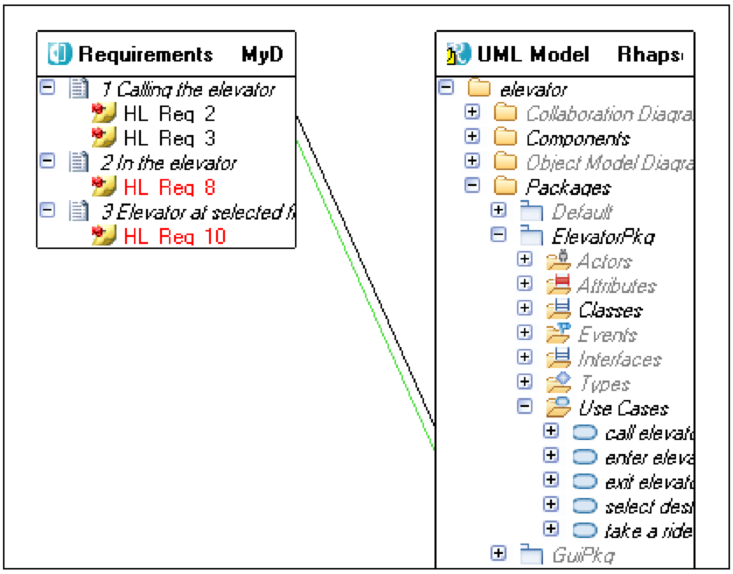
Creating Rhapsody anchors is an old means for creating traceability links; this is always useable. See the Advanced tab of the **Add high level** box.

See the *Rhapsody User Guide* for more information.

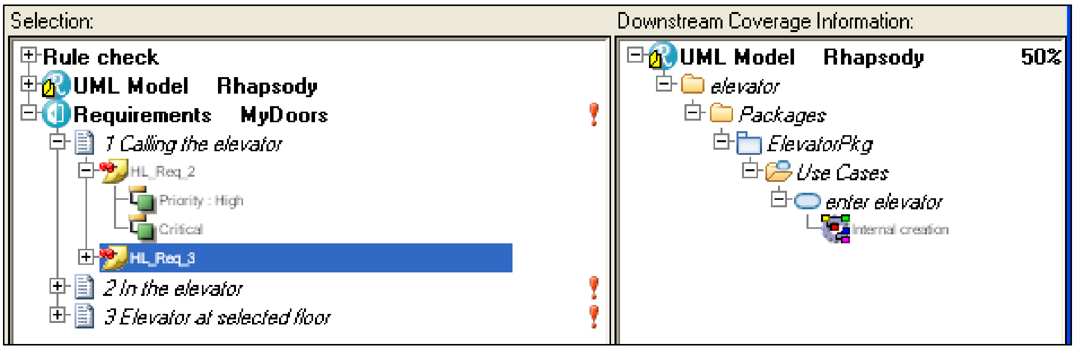
Links Created from Rhapsody Gateway

Some coverage links can be created from Rhapsody Gateway using the **Graphical Mode** view or the **Link Details** view. This is interesting when we do not want or cannot modify some documents. See the **Adding Covering Links** from the Rhapsody Gateway *User Guide* document.

The following picture displays a created covering link. It is represented with a green line.



Covering links are presented in the Rhapsody Gateway main window with a reference attribute "Internal creation", see following picture.



Selection:

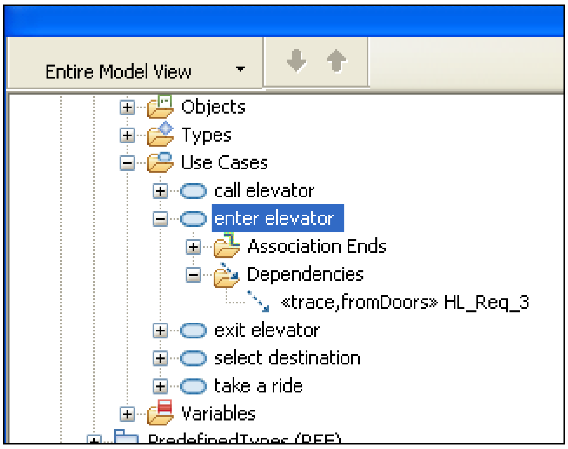
- Rule check
- UML Model Rhapsody
- Requirements MyDoors
  - 1 Calling the elevator
    - HL_Req_2
      - Priority : High
      - Critical
    - HL_Req_3
  - 2 In the elevator
  - 3 Elevator at selected floor

Downstream Coverage Information:

- UML Model Rhapsody 50%
  - elevator
    - Packages
      - ElevatorPkg
        - Use Cases
          - enter elevator
            - Internal creation

To insert externally created links into Rhapsody, click **Tools** > **Add high level requirements**. If **Create Dependencies instead of Anchors** is selected in the **Add high level requirements** option, these links are imported in Rhapsody as **Dependencies**.

This is shown in the following Rhapsody screenshot.



Entire Model View

- Objects
- Types
- Use Cases
  - call elevator
  - enter elevator
    - Association Ends
    - Dependencies
      - «trace,fromDoors» HL_Req_3
  - exit elevator
  - select destination
  - take a ride
- Variables
- PredefinedTypes (REF)

Exporting to DOORS

Rhapsody Gateway allows the upload into DOORS of lower level information with the traceability related to DOORS requirements. See the *Coupling DOORS* note for information about DOORS-Rhapsody Gateway interface.

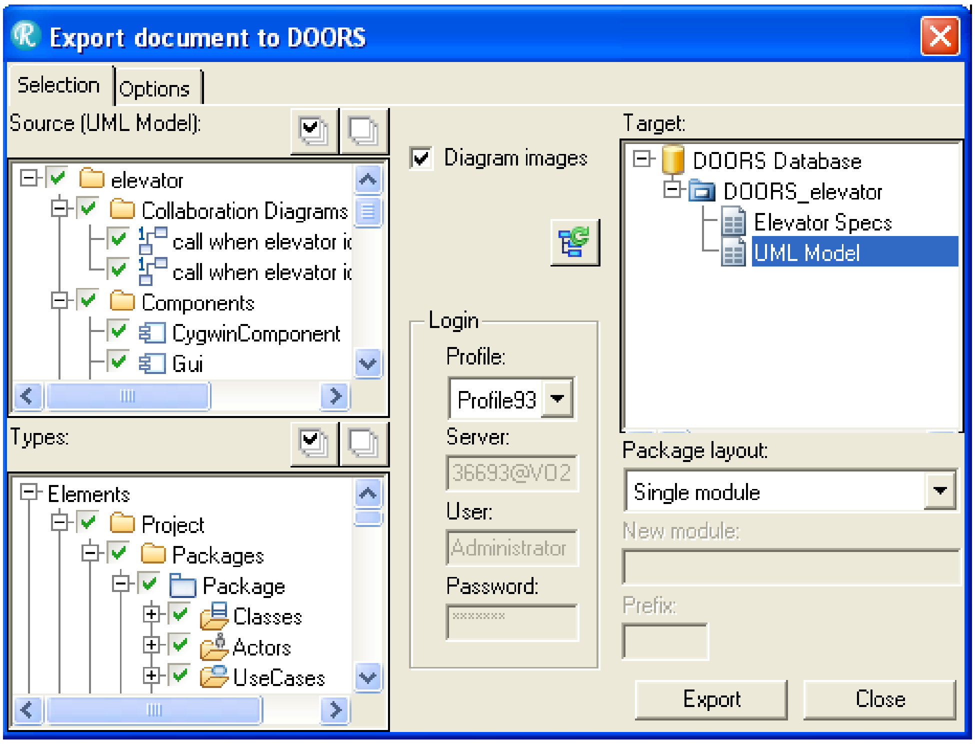
This section describes the upload process of elements into DOORS.

Creating Rhapsody Elements into DOORS

You can upload the model into DOORS with or without diagrams. Obviously, the diagrams need to be captured before being exported.

1. Select the UML Model from Rhapsody Gateway then choose **Tools** > **Export document to DOORS**.

The **Export document to DOORS** dialog box appears.



Export document to DOORS

☑ Diagram images

Profile: Profile93

Server: 36693@V02

User: Administrator

Password: *******

Package layout: Single module

New module:

Prefix:

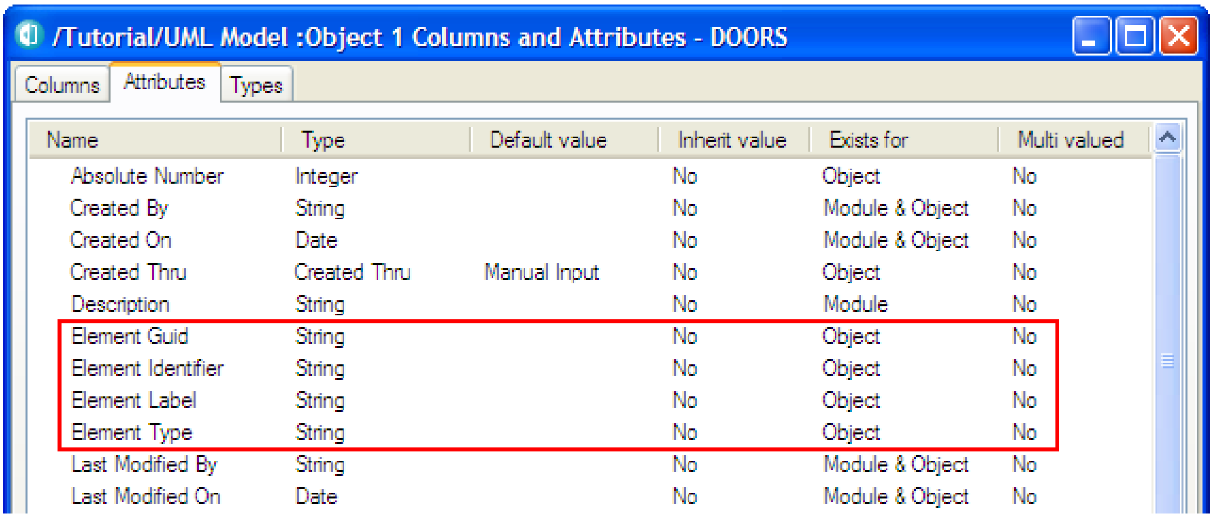
From this box, select the elements to export:

- The **Source** list displays the current model hierarchy. Select the parts of the hierarchy that are to be uploaded.
- The **Types** list displays a hierarchic ordering of element types available in the source document type.

Note

Please refer to the “Export to DOORS” section of the *DOORS Coupling Notes* for further information on the other options of this Export window (**Diagram images**, **Server**, **Package layout**, etc. and options in the **Options** tab).

2. Select a location in the DOORS database in the **Target** view.
3. Enter a module name. The default name is the name of the document defined in the Rhapsody Gateway project editor for the Rhapsody model.
4. Enter a **Prefix** if you want the created DOORS objects to have a particular prefix.
5. Click **Export**.
   Rhapsody Gateway dialogs with DOORS to check what needs to be updated. An information window appears. If it is the first upload, you will have all the Rhapsody elements presented as “New elements”.
6. Click **Export** to launch the upload of the information into the DOORS database. (See next section to have detailed information on tags export) Use **Cancel** to close the dialog box without any action in DOORS.
7. Open the **Columns and Attributes** window from DOORS to visualize the upload result in DOORS. Rhapsody Gateway creates four attributes for the imported module:
   - Element Guid
   - Element Identifier
   - Element Label
   - Element Type



/Tutorial/UML Model :Object 1 Columns and Attributes - DOORS
Columns
Attributes
Types

Name	Type	Default value	Inherit value	Exists for	Multi valued
Absolute Number	Integer		No	Object	No
Created By	String		No	Module & Object	No
Created On	Date		No	Module & Object	No
Created Thru	Created Thru	Manual Input	No	Object	No
Description	String		No	Module	No
Element Guid	String		No	Object	No
Element Identifier	String		No	Object	No
Element Label	String		No	Object	No
Element Type	String		No	Object	No
Last Modified By	String		No	Module & Object	No
Last Modified On	Date		No	Module & Object	No

These attributes are not supposed to be managed by the user; they are used by Rhapsody Gateway to manage objects.

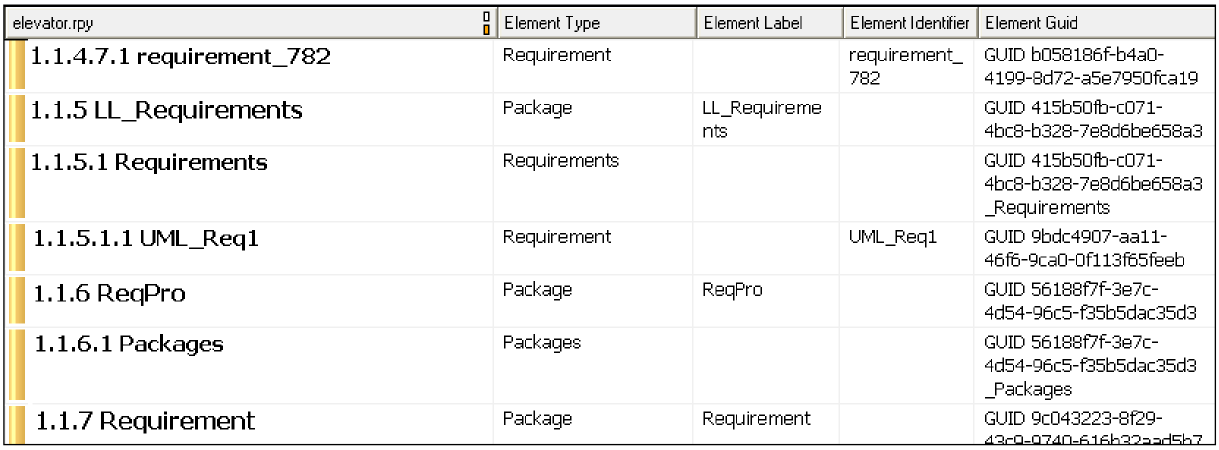
The **Specification** fields in Rhapsody are uploaded into DOORS as **Object texts** and diagrams are also inserted as **Object texts** as well.

Important note

Rhapsody is considered to be the reference for Rhapsody information. If you change in DOORS the **Object text** uploaded from a Rhapsody **Specification** field, your modification will be erased by the next reload action and replaced by the Rhapsody Specification.

Tip: Modify a Rhapsody element in Rhapsody, not in DOORS.

8. Open the **Standard** view to visualize the contents of the upload.



elevator.rpy	Element Type	Element Label	Element Identifier	Element Guid
1.1.4.7.1 requirement_782	Requirement		requirement_782	GUID b058186f-b4a0-4199-8d72-a5e7950fca19
1.1.5 LL_Requirements	Package	LL_Requirements		GUID 415b50fb-c071-4bc8-b328-7e8d6be658a3
1.1.5.1 Requirements	Requirements			GUID 415b50fb-c071-4bc8-b328-7e8d6be658a3_Requirements
1.1.5.1.1 UML_Req1	Requirement		UML_Req1	GUID 9bdc4907-aa11-46f6-9ca0-0f113f65feeb
1.1.6 ReqPro	Package	ReqPro		GUID 56188f7f-3e7c-4d54-96c5-f35b5dac35d3
1.1.6.1 Packages	Packages			GUID 56188f7f-3e7c-4d54-96c5-f35b5dac35d3_Packages
1.1.7 Requirement	Package	Requirement		GUID 9c043223-8f29-43c9-9740-616b32aad5b7

Note

You can add your own attributes in this created DOORS module and create links from or to this created module. They will not be modified by Rhapsody Gateway.

Note

Rhapsody **Tags** are created as DOORS **attributes**, as described in the next section.

Managing, Capturing and Exporting Tags

All Rhapsody tags are created as DOORS attributes for the imported module.

Rhapsody tags can be captured and taken into account in Rhapsody Gateway and as a result be included in the third party tool.

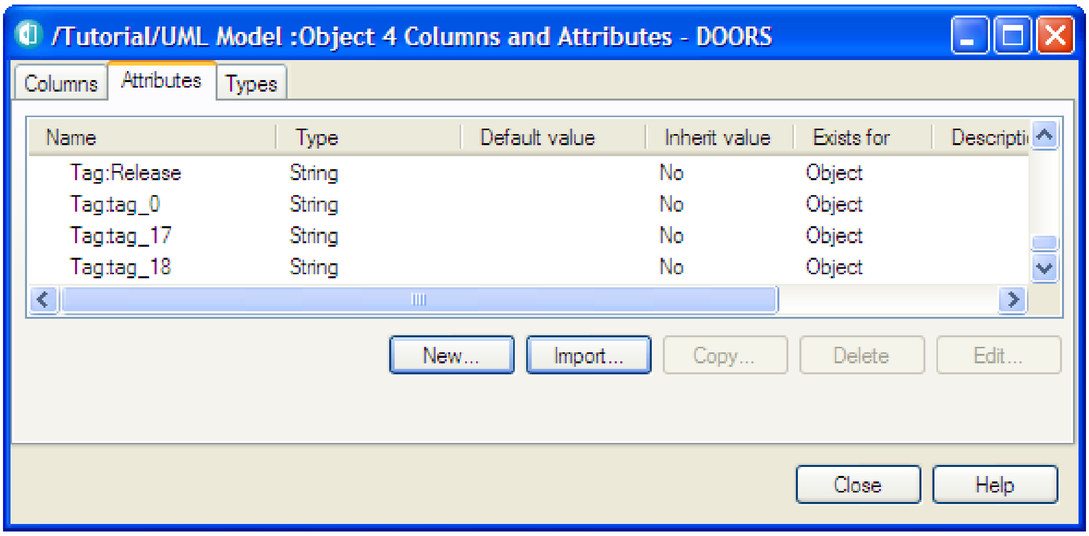
DOORS supports the generic tags export, the specific tags export, the enumeration export and the multi-values export if it is enumeration.

You can export tags from Rhapsody to DOORS, whether a type in Rhapsody Gateway which contains the corresponding attributes is created. See *Capturing Generic and Specific Attributes* to have detailed information.

Exporting Generic Tags

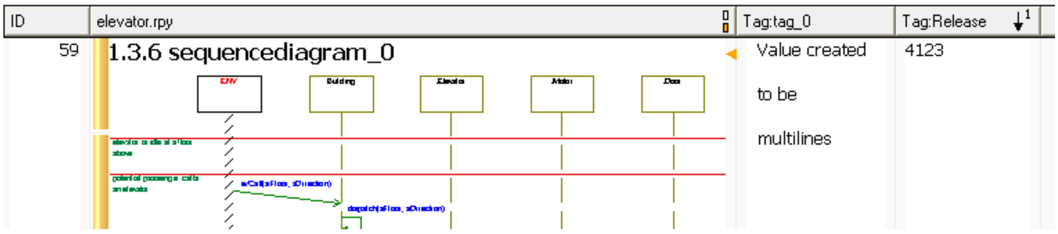
All the Rhapsody tags are created as DOORS attributes for the imported module. Each generic attribute is named `Tag:<tag name>`.

The example below shows an addition creation.



Name	Type	Default value	Inherit value	Exists for	Descripti
Tag:Release	String		No	Object	
Tag:tag_0	String		No	Object	
Tag:tag_17	String		No	Object	
Tag:tag_18	String		No	Object	

These attributes are completed with the value captured for attributes by Rhapsody Gateway.



ID	elevator.rpy	Tag:tag_0	Tag:Release
59	1.3.6 sequencediagram_0	Value created to be multilines	4123

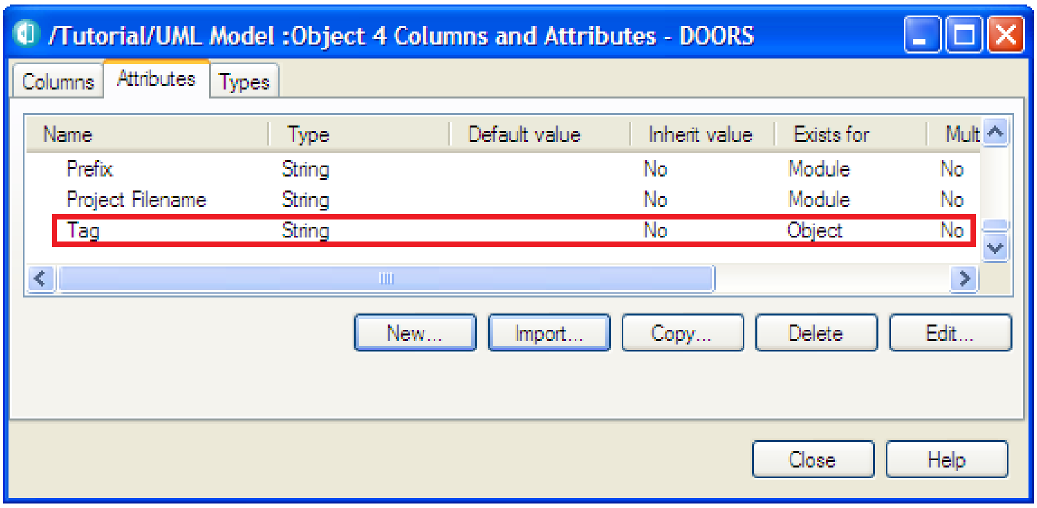
Note

Tags can be defined in Rhapsody Gateway as Boolean attributes. Tags may be part of the Rhapsody Gateway customization for a particular customer process and then be created in Doors as Boolean. Default configuration considers tags to be valued. If tags are captured from Rhapsody but none of them has a value defined, the attribute will not be created in DOORS. As soon as at least one of the tags has a defined value, the attribute is created in DOORS and the value is filled for the corresponding object.

Exporting Specific Tags

If some specific attributes are captured from Rhapsody Gateway, only one attribute is created. This tag is named **Tag**.

The example below shows the addition of an attribute.



/Tutorial/UML Model :Object 4 Columns and Attributes - DOORS

Name	Type	Default value	Inherit value	Exists for	Mult
Prefix	String		No	Module	No
Project Filename	String		No	Module	No
Tag	String		No	Object	No

Traceability and Links

Rhapsody Gateway defines several ways to capture traceability information through **References** and **Links** in the type definition. See the *Customization Guide* and the *User Guide* to learn more about these concepts.

The default configuration for Rhapsody assumes:

- A definition of References captures the traceability through the detection of Rhapsody dependencies using stereotypes such as trace, satisfy or verify. This type of element is called a Reference.
- A definition of Links captures in the Rhapsody model the internal relationships and dependencies with stereotypes other than trace, satisfy and verify.

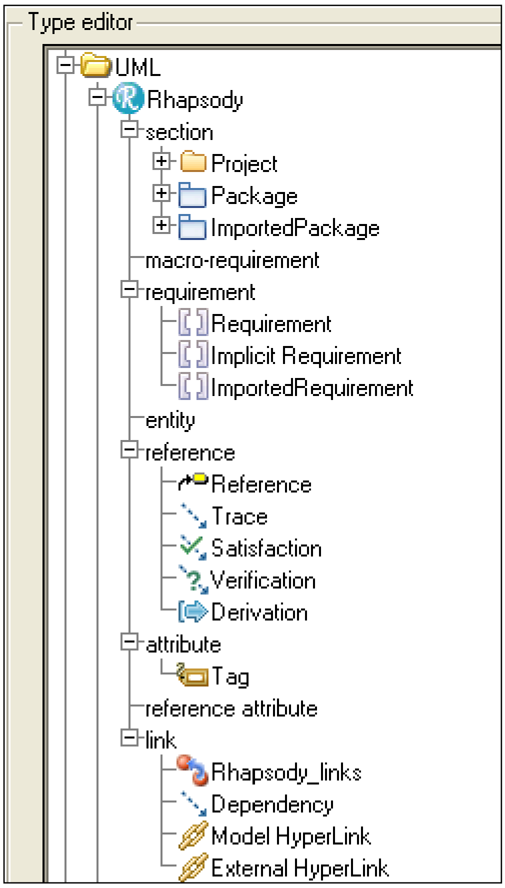
One link module is created in DOORS for each kind of **Reference or Link** defined in Rhapsody Gateway.

Hyperlinks can be captured in Rhapsody and then be exported to DOORS. Depending on the various types of file that can be referenced, two hyperlink types are defined:

- Model HyperLink: for Rhapsody hyperlinks pointing to a model element
- External HyperLink: for Rhapsody hyperlinks pointing to anything else (**Error! Hyperlink reference not valid.**, http://, free text identifier, etc.)

See the following figure to compare the link creation.

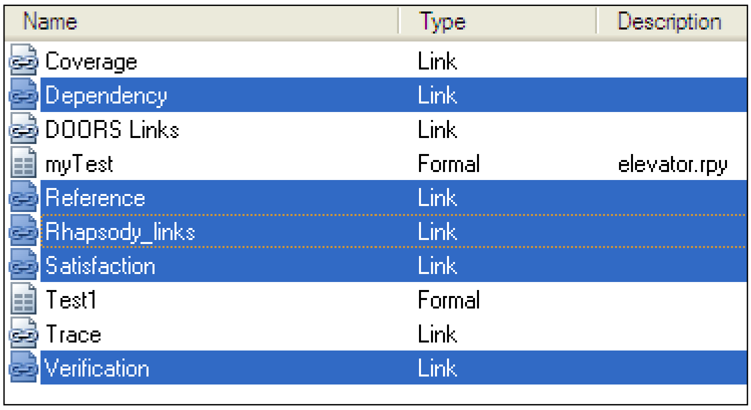
Type definition in Rhapsody Gateway



Type editor

- UML
  - Rhapsody
    - section
      - Project
      - Package
      - ImportedPackage
    - macro-requirement
    - requirement
      - Requirement
      - Implicit Requirement
      - ImportedRequirement
    - entity
    - reference
      - Reference
      - Trace
      - Satisfaction
      - Verification
      - Derivation
    - attribute
      - Tag
    - reference attribute
    - link
      - Rhapsody_links
      - Dependency
      - Model HyperLink
      - External HyperLink

Corresponding created link modules in the DOORS database



Name	Type	Description
Coverage	Link	
Dependency	Link	
DOORS Links	Link	
myTest	Formal	elevator.rpy
Reference	Link	
Rhapsody_links	Link	
Satisfaction	Link	
Test1	Formal	
Trace	Link	
Verification	Link	

Exporting to RequisitePro

How to upload actions into RequisitePro is fully described in the *Coupling RequisitePro* note, installed in the `doc` subdirectory below the installation directory of Rhapsody Gateway.

Support of Profiles

Profiles are used to improve UML models. Some Rhapsody profiles are provided in the form of a type integrated into Rhapsody Gateway such as AUTOSAR, FunctionalC, SysML, etc. Additional types arising from user profiles can also be inserted.

Rhapsody 7.2 manages profiles compliant with SysML and DoDAF standards. These profiles are supported by Rhapsody Gateway. All the concepts described in the previous sections apply. Some concepts have been added to create a requirements-oriented management of these processes.

This section provides information on the following topics:

- Rhapsody Gateway and SysML
- Rhapsody Gateway and DoDaF
- Considering Other Profiles

Rhapsody Gateway and SysML

The dedicated type for SysML support is **Rhapsody SysML**. It analyzes the SysML model and presents it as other Rhapsody models but performs an additional analysis of **dependencies**.

The SysML standard (i.e. SysML specification v0.9 released January 2005) takes into account particular relationships for requirement diagrams such as:

- Satisfy
- Verify
- Trace

For these dependencies, one of the constraints is *“The supplier must be an element stereotyped by <<requirement>>”*.

Rhapsody allows you to create dependencies, including the ones mentioned, between any kind of elements. The constraints are defined at process level and are not imposed by Rhapsody itself.

However, as these dependencies represent important requirement traceability information, a dedicated analysis is performed by Rhapsody Gateway:

- **Satisfy**, **Verify** and **Trace** dependencies are defined as ‘References’ in the **Rhapsody SysML** type. This means that they will be captured as part of the traceability (coverage) information.

- All other dependencies are defined as ‘Links’ in the Rhapsody SysML type. Rhapsody Gateway presents the information between objects but these links are not part of the traceability (it is not their role).

If the user defines a **Satisfy**, **Verify** or **Trace** dependency between an object and another object which is not a requirement in Rhapsody, Rhapsody Gateway will automatically create an element beneath the referenced object. This element has the same name and is defined as an “implicit requirement”.

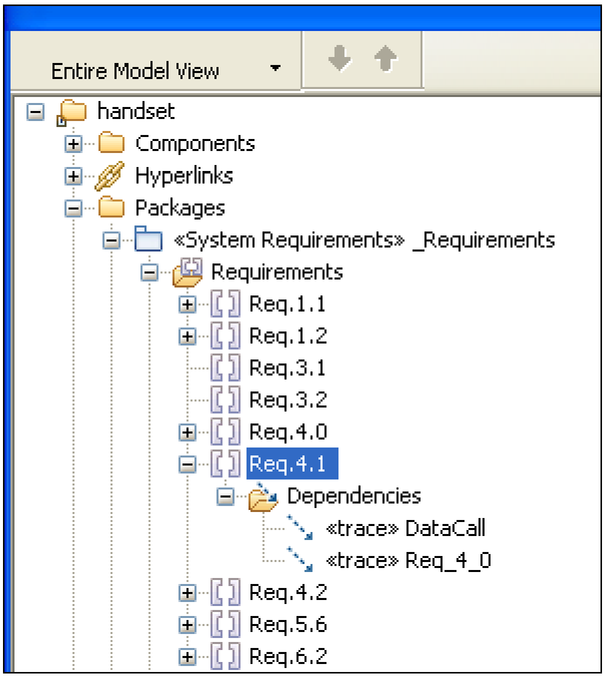
The example below comes from the Handset delivered in Rhapsody. As an example, the figure below shows the Rhapsody requirement `Req.4.1`. This requirement contains two dependencies:

- one derives `Req_4_0` (Req.4.0) which is a requirement.
- the other one traces `DataCall`, which is not a requirement but a Use Case.

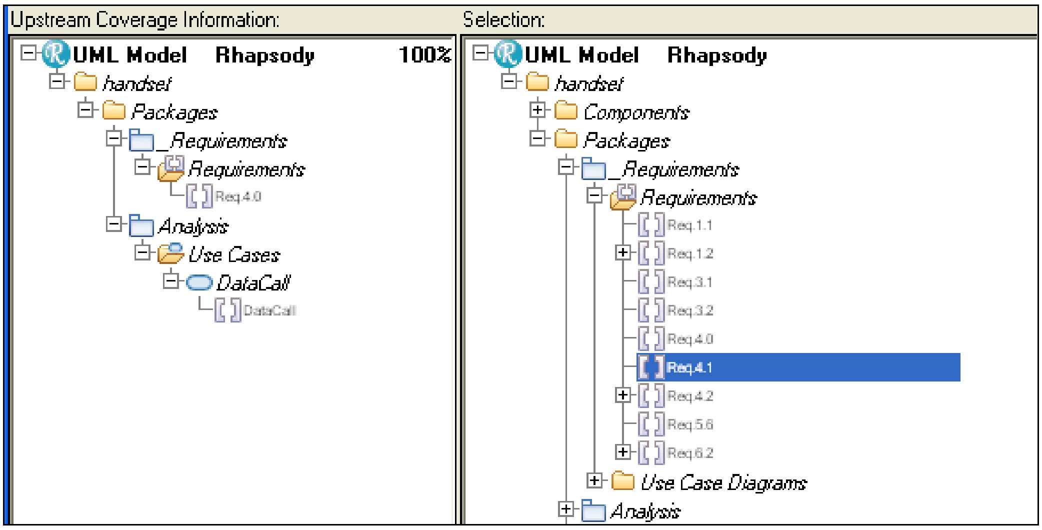
Change the stereotype of Req_4_0 into “trace”.

Having a <<trace>> dependency between a requirement and a Use Case is allowed by Rhapsody.

In a requirements-oriented analysis like the one performed by Rhapsody Gateway, this Use Case would be considered as a requirement which is traced (covered) by `Req.4.1.`

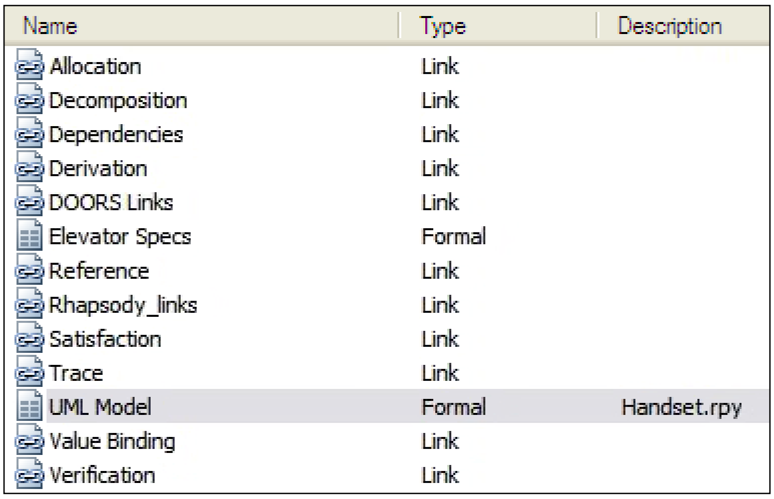


So Rhapsody Gateway will automatically create an additional element in its tree (but not in Rhapsody) called ‘Implicit requirement’ with the same name as the traced object. The figure below shows Req.4.1 tracing Req.4.0, but also an implicit requirement Datacall, inserted beneath the DataCall Use Case of the Rhapsody model.



Note

In the case of a DOORS export, the support of these dependencies described above for SysML will result in the creation of a one link module for each dependency.



Name	Type	Description
Allocation	Link	
Decomposition	Link	
Dependencies	Link	
Derivation	Link	
DOORS Links	Link	
Elevator Specs	Formal	
Reference	Link	
Rhapsody_links	Link	
Satisfaction	Link	
Trace	Link	
UML Model	Formal	Handset.rpy
Value Binding	Link	
Verification	Link	

Rhapsody Gateway and DoDAF

Rhapsody Gateway also supports Rhapsody models created according to the DoDAF standard and using the DoDAF stereotype. The dedicated type for this support is **Rhapsody DoDAF**.

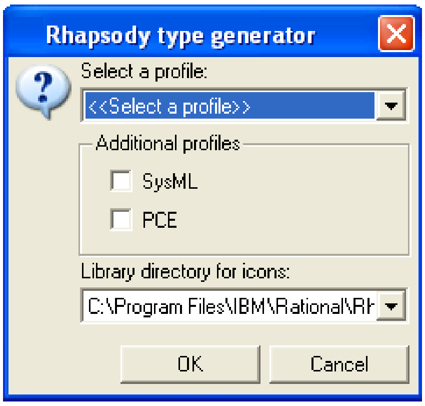
The concept of implicit requirements which are automatically created by Rhapsody Gateway also exists for DoDAF when the dependencies use the **Trace** stereotype.

Considering Other Profiles

Rhapsody Gateway recognizes profiles. But all the profiles created by users cannot be found in Rhapsody Gateway as types. An option enables the generation of types which match the profiles applied to Rhapsody models, To better reflect information coming from Rhapsody.

To create new types matching profiles from Rhapsody, follow these steps:

1. Select the UML Model of the traceability project in Rhapsody Gateway.
2. Select the **Generate types for this model** option from the **Tools** menu.
3. Select the profiles to generate from the open dialog box.



Rhapsody type generator

Select a profile: <<Select a profile>>

Additional profiles

□ SysML

□ PCE

Library directory for icons: C:\Program Files\IBM\Rational\Rh

Select the main profile for the type from the combo-box.

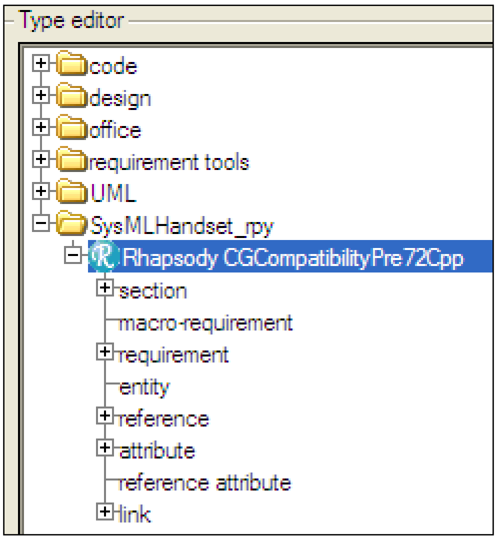
You can also select one or more additional profiles if some are suggested.

Select the directory to save icons coming from the profiles and which will be used in the new created type.

Note:

If the UML Model does not contain profiles the following message appears.

4. Open the type editor 🔍. A new type has been inserted into the Type editor.

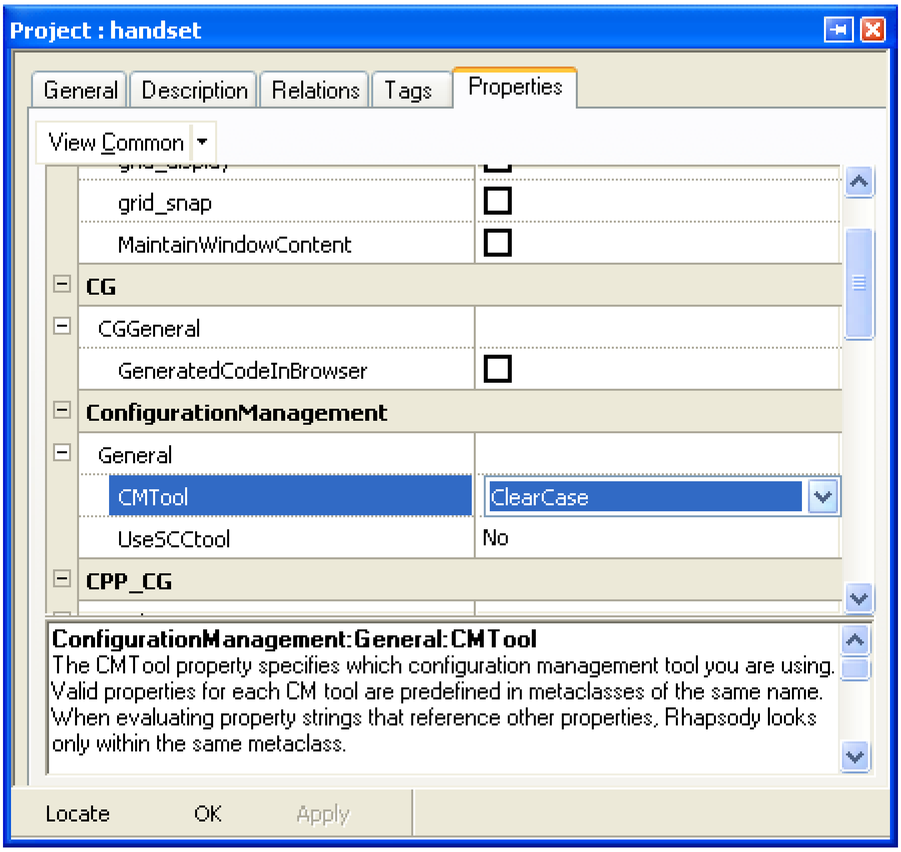


5. The corresponding Rhapsody document type in the Project configuration has automatically changed with the newly generated type. Launch the **Reload** feature on the model to re-execute the analysis of the model with the new type.

Rhapsody CM Capabilities

Rhapsody integrates the capability to manage the model configuration and provides a front-end to a Configuration Management (CM) tool (PVCS, Clearcase and SourceIntegrity).

Refer to Rhapsody documentation for all these features.



grid_snap	□
MaintainWindowContent	□
CG	
CGGeneral	
GeneratedCodeInBrowser	□
ConfigurationManagement	
General	
CMTool	ClearCase
UseSCCtool	No
CPP_CG	

ConfigurationManagement:General:CMTool
The CMTool property specifies which configuration management tool you are using. Valid properties for each CM tool are predefined in metaclasses of the same name. When evaluating property strings that reference other properties, Rhapsody looks only within the same metaclass.

This section describes how to include Rhapsody Gateway projects in the Rhapsody-centric Configuration Management (CM) process.

If we consider a Rhapsody Gateway project with a Rhapsody model tracing a Word document, the typical steps are as follows:

1. Create a Rhapsody Gateway project from Rhapsody. The Rhapsody Gateway project files are already saved in the `_rpy` sub-directory of your Rhapsody project directory.

2. Setup Rhapsody Gateway project properties (project configuration, types of analysis, filters, etc.). All the files corresponding to these definitions are automatically saved by Rhapsody Gateway.

3. In Rhapsody, add Rhapsody Gateway project files as **controlled files** in the Rhapsody browser.

   The files to consider are:

   - The `<project name>.rqtf` file—This file corresponds to the project definition: it always exists.

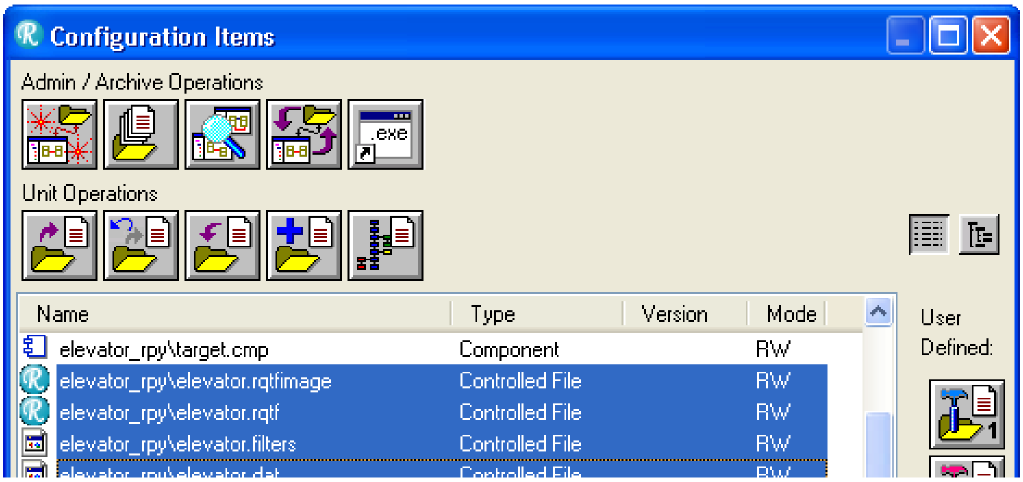
   - The files with the extension `.types` from the `_rpy` subdirectory.—These files define the types of analysis available for your Rhapsody Gateway project. If the Rhapsody Gateway project only uses default types, no `.types` file is available in the directory.

   - The `<project name>.filters` file—This file corresponds to the filters defined for the project. If no filter is defined for the project no .filter file is available in the directory.

   - The `<project name>.dat` file—This file corresponds to additional information entered for snapshots. If you did not use this feature, you will have no .dat file.

   - The `<project name>.rqtfimage` file—This file contains the analysis results. It always exists in the Rhapsody Gateway project directory as soon as the first analysis has been performed. This first analysis is performed when you create the Rhapsody Gateway project from Rhapsody.

4. Open the Configuration Items to add all these files to the archive.



Name	Type	Version	Mode
elevator_rpy\target.cmp	Component		RW
elevator_rpy\elevator.rqtfimage	Controlled File		RW
elevator_rpy\elevator.rqtf	Controlled File		RW
elevator_rpy\elevator.filters	Controlled File		RW
elevator_rpy\elevator.dat	Controlled File		RW

   Select the files then click **Add the selected items to Archive** option.

5. From the Configuration Items check in all these files. The configuration management process consists in classical check-in / check-out operations.

IMPORTANT

It is strongly recommended to consider the entire Rhapsody Gateway project in these operations and not only as a subset.

It is essential to consider the source documents in a consistent approach. Users have to include analyzed documents such as Word documents in their CM actions and make available not only the Rhapsody Gateway files but also the analyzed files. These files need to be inserted in the same subdirectory as the Rhapsody Gateway project (the `_rpy` subdirectory).

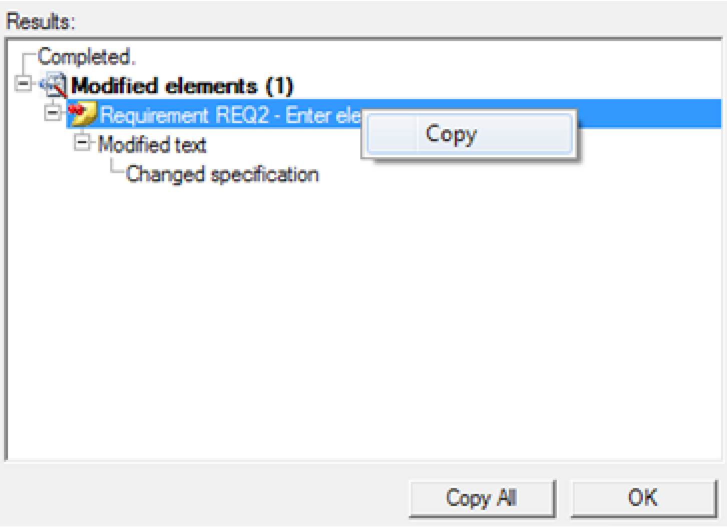
As for other documents, the Rhapsody Gateway project will expect a Rhapsody project with a given name. When the Rhapsody Gateway project is created from a **MyModel** Rhapsody model, Rhapsody Gateway analyzes the `MyModel.rpy` file. All users will have to work with a Rhapsody local project called **MyModel**. If a local project is called **NewName**, the local file will be `NewName.rpy`. Rhapsody Gateway will no longer be able to analyze the `MyModel.rpy` file defined in its project configuration and will cause an error.

Troubleshootings

Preserving Add High Level Requirements Logs

Add high level requirements provides different ways to keep logs of modification performed and messages.

The status dialog box provides a Copy All button and a right-click menu to copy textual information into the clipboard:



Results:

- Completed.
- Modified elements (1)
  - Requirement REQ2 - Enter ele
    - Modified text
      - Changed specification

The status is automatically printed in textual form to the Rhapsody Gateway project log file while Rhapsody Gateway is running in batch mode.

It is also possible to set the following environment variables to preserve automatically the logs in XML format.

- SYNC_LOG_XMLFILE=<logfile.xml> where the XML filename is absolute or relative to the Rhapsody Gateway project file.
- SYNC_LOG_BACKUP=1 to preserve previous XML log files with a date/time suffix.