

IBM® Rational® Rhapsody® Gateway Add On



Rhapsody Coupling Notes

Rhapsody[®]

**IBM[®] Rational[®] Rhapsody[®]
Gateway Add On**

Rhapsody Coupling Notes



License Agreement

No part of this publication may be reproduced, transmitted, stored in a retrieval system, nor translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of the copyright owner.

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

Reqtify is a registered trademark of Dassault Systèmes or its affiliates in the US and/or other countries.

Rhapsody Gateway, IBM, the IBM logo, DOORS and Rhapsody are trademarks or registered trademarks of IBM Corporation.

All other product or company names mentioned herein may be trademarks or registered trademarks of their respective owners.

© Copyright 2001-2011 Dassault Systèmes. All rights reserved.

Contents

Contents.....	5
Preface	7
Rhapsody Analysis	9
Importing Requirements.....	9
Rhapsody Analysis Type Parameters.....	10
Changing the Language of Analysis	11
Activating or Deactivating the Capture of Diagrams.....	11
Selecting Packages	12
Adding stereotypes.....	12
Rhapsody Options	12
Displaying Labels or Elements Names	13
Selecting Multi-Project Behavior.....	14
Customizing Analysis Type	14
High Level Requirements Addition	15
Requirements Creation into Rhapsody.....	15
High Level Requirements Synchronization Process	22
New Requirements	22
Modified Requirements.....	22
Deleted Requirements	22
Reaching a Requirement	23
Attributes / Rhapsody Tags Relationships.....	25
Representation of Requirements Attributes	25
Capturing Rhapsody Tags	26
General Attributes Capture	26
Capturing Specific Attributes.....	27
Requirements Traceability	29
Using Dependencies.....	29
Using Rhapsody Anchors.....	30
Links Created from Rhapsody Gateway	30
Exporting to DOORS.....	33
Creating Rhapsody Elements into DOORS.....	33
Tags Management	36
Traceability and Links.....	37
Export to RequisitePro	39
Support of Profiles	41

Contents

Rhapsody Gateway and SysML	41
Rhapsody Gateway and DoDAF	43
Considering Other Profiles	44
Rhapsody CM Capabilities.....	47

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 Manual*.

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

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 in order to create a traceability project and some alternatives to configure the project.

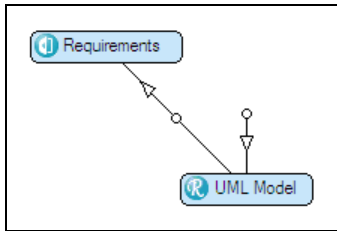
Importing Requirements

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 Manual* for more details).

1. Click on the **Add a document**  button 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 the **Add a cover**  button 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.



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.

Rhapsody Analysis Type Parameters

When you insert a Rhapsody model into the Rhapsody Gateway project editor, you are able to modify the default parameters. A **Variable** drop down list allows you to access the available variables that can be used for the parameters of a selected Rhapsody document. Three parameters are available for each Rhapsody analysis type:

- ◆ Language
- ◆ Capture diagrams

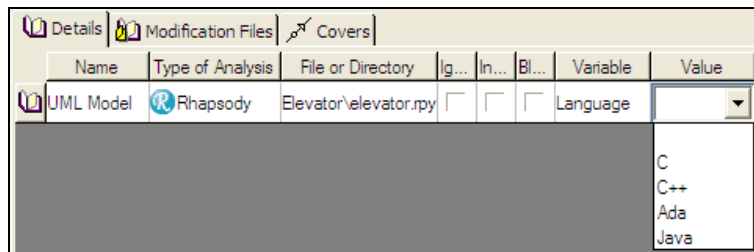
- ◆ Package
- ◆ With stereotypes applied

Changing the Language of Analysis

If you do not modify the value of **Language** variable, its default value is C++.

To change this default setting, follow these steps:

1. Select the Rhapsody model in the Project Configuration.
2. Click **Language** in the **Variable** drop down list.
3. Select the language in the **Value** drop down list. Available values are C, C++, Ada and Java languages.

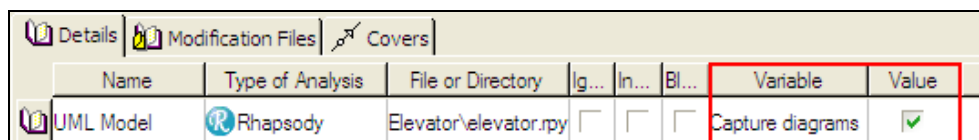


Activating or Deactivating the Capture of Diagrams

The **Capture diagrams** variable allows you to capture vectorized diagrams from Rhapsody elements such as sequence diagrams. You can for instance export these diagrams into DOORS.

To choose whether or not to capture the Rhapsody diagrams, follow these steps:

1. Select the Rhapsody model in the Project editor.
2. Select **Capture diagrams** in the **Variable** drop down list.
3. Activate or deactivate the check box in the **Value** field.



Note



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

Selecting Packages

The **Package** variable allows you to reduce analysis from all the models to only one package and its sub-packages.

To select one specific package, follow these steps:

1. Select the Rhapsody model in the Project editor.
2. Select **Package** in the **Variable** drop down list.
3. Type the name of the package in the **Value** field.

Details Modification Files Covers								
	Name	Type of Analysis	File or Directory	Ig...	In...	Bl...	Variable	Value
	UML Model	 Rhapsody	Elevator\elelevator.py	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Package	ElevatorPkg

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.

Adding stereotypes

The **With stereotypes applied** variable allows you to complete the XML intermediate file with stereotypes applied on each model element. This information is interesting when using an XML customized type. To use this variable, activate or deactivate the check box in the **Value** field.

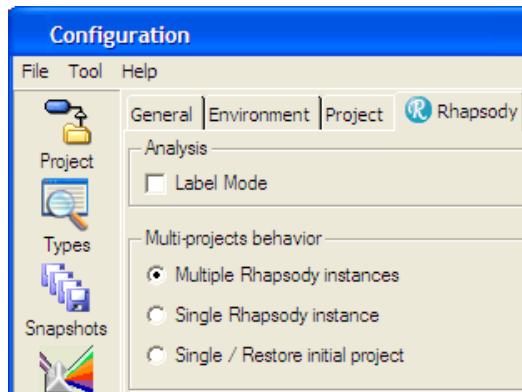
Rhapsody Options

Some application options can be set from Rhapsody Gateway.

To access the Rhapsody options from Rhapsody Gateway, follow these steps:

1. Select **Options** in the **Tools** menu.

2. In the window which opens, select the **Rhapsody** tab:



This tab contains two option groups described below.

Displaying Labels or Elements Names

Rhapsody allows you to display either labels or element names, depending on a model browser setting (Label Mode).

To change this setting, follow these steps:

1. Check the **Label Mode** option to display labels defined in the model. When the **Label Mode** option is unchecked, element names are displayed by default.
2. Click **OK** and reload the model (right click on UML Model and select the **Reload** option) to take into account the changes, so the analysis results provide the requested information.

Note

Rhapsody Gateway allows (advanced) users to 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 by selecting the **Reload All** option in the **File** menu.

Selecting Multi-Project Behavior

Rhapsody allows you to provide GUI for the Rhapsody connection mode setting.

The following options are available in the Multi-project behavior:

- ◆ **Multiple Rhapsody instances**—Select this option if you want a temporary Rhapsody instance to be created in order to convert unloaded Rhapsody projects. This option is checked by default unless another setting is specified.
- ◆ **Single Rhapsody instance**—Select this option if you want every project to be loaded in the current instance of Rhapsody.
- ◆ **Single / Restore initial project**—Select this option if you want every project to be loaded in the current instance of Rhapsody and to have the original project restored at the end of the operation.

Customizing Analysis Type

If the default definition of a type does not comply with your needs, you must create a customized type of analysis. This makes it is possible to replace the default definition for requirement identifiers or for attributes with a customized definition.

The procedure consists of duplicating the document Rhapsody type, modifying the type, then associating this new type with the document.

See the *Customization Guide* for more information on this subject.

Note

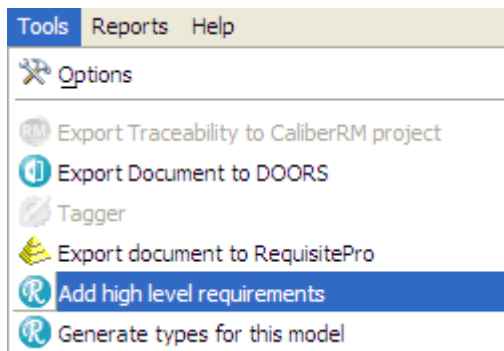
Customizing type is an advanced feature of Rhapsody Gateway. It is uncommon to need to redefine Rhapsody types because they are directly provided by Rhapsody analysis. See the **Considering Other Profiles** section of this document to learn more.

High Level Requirements Addition

The traceability requirements coming from the third level document need to be re-entered into Rhapsody to be covered by model elements. This section develops the corresponding process.

Requirements Creation into Rhapsody

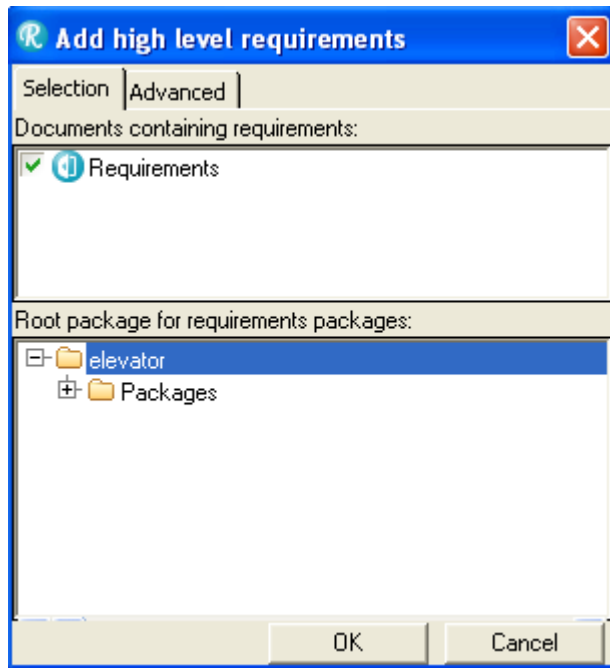
When a Rhapsody model is selected, the tool interface includes an available item in the Rhapsody Gateway **Tools** menu, as shown in the following example.



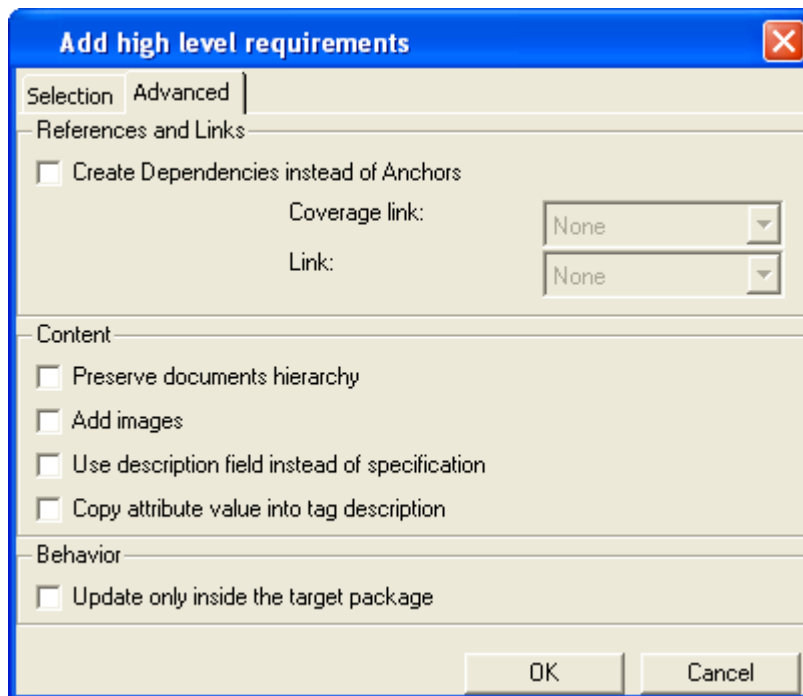
In order to perform the requirements addition, follow these steps:

1. Once requirements have been captured from the high level document, select the root of a Rhapsody-based document, then select **Add high level requirements** from the **Tools** menu.

The following dialog window opens:



2. Now, select:
 - ◆ The requirements you want to import, if you have several documents covered by the Rhapsody model.
 - ◆ The location of the requirements package in the Rhapsody model tree. If it is the first time, select the 'parent' in the tree, then the selected location of the package will be proposed by default for next exports.
3. An **Advanced** pane allows you to specify how the links will be created in the Rhapsody model and how the document's hierarchy can be materialized into Rhapsody.



- **Create Dependencies instead of Anchors** option—indicates whether or not you want to create links as dependencies or as anchors in Rhapsody. By default, this option is unchecked.

When the **Create Dependencies instead of Anchors** option is checked, additional list boxes **Coverage Link** and **Link** allow the user to select the stereotypes of the links created as dependencies.

- **Preserve documents hierarchy** option—indicates whether or not you want to keep the hierarchy of the documents 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 unchecked.
- **Add images** option—allows you to add images into Rhapsody such as those that come from DOORS. By default, this option is unchecked.
- **Use description field instead of specification** option—enables the display of text from Rhapsody Gateway into the **Description** pane instead of the **Specification** field. The **Description** pane supports unicode characters (such as Chinese characters, Greek characters, etc.). By default, this option is unchecked.
- **Copy attribute value into tag description** option—enables the copy of the tag value of high level requirements into the tag description field.
- **Update only inside the target package** option—enables you to control the scope of the requirements synchronization. Rhapsody Gateway performs a global synchronization when this option is unchecked. 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 checked when the synchronization must be performed locally. The local synchronization behavior might be useful when synchronizing the documents one by one.

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, or if you select a new location for export, you can duplicate requirements in Rhapsody.

4. At the end of the adding operation, you will see a report message, such as the following.



5. 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_eleva

General Description Relations Tags Properties

Name: HL_Req_2

Stereotype: fromMyDoors

Type: Requirement

ID: HL_Req_2

Defined in: _1_Calling_the_elevator

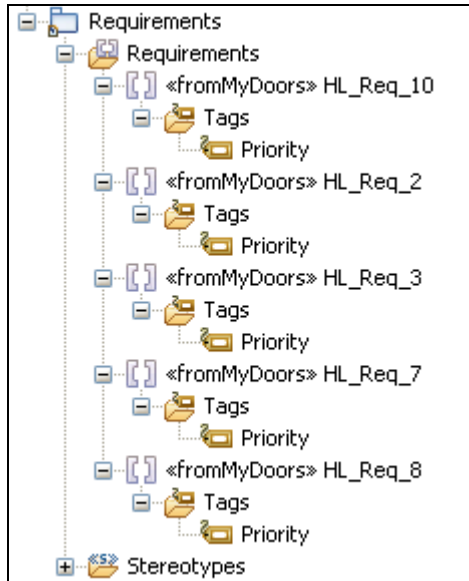
Specification: A potential passenger can be on any of the floors and can call an elevator by pressing either the up or down button to call the elevator

Below is an example of the creation resulting from Rhapsody requirements taken 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	

Corresponding View of the Rhapsody tree

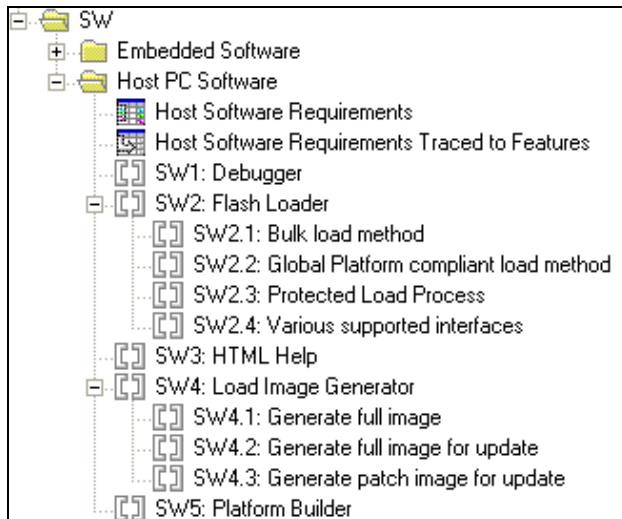


Note

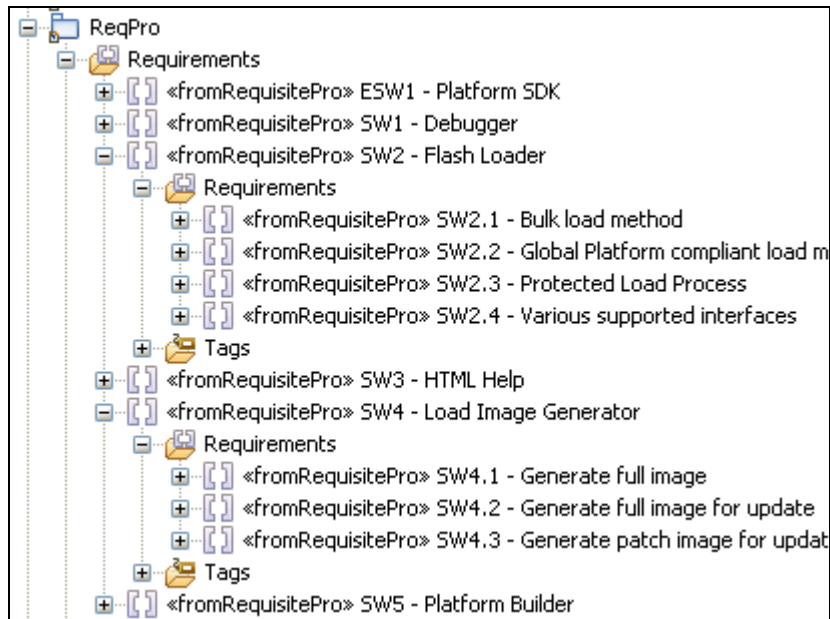
If the **Preserve documents hierarchy** option is checked, some packages are created to keep the hierarchy of the high level document in Rhapsody.

Below is an example of the creation resulting from Rhapsody requirements taken from RequisitePro:

View of the RequisitePro tree

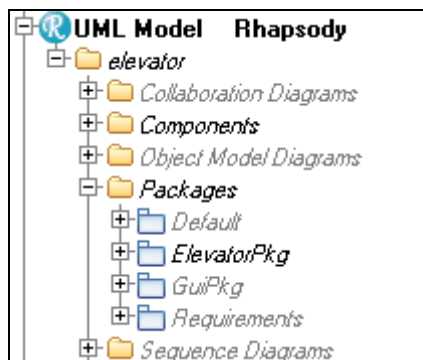


Corresponding View of the Rhapsody tree



Important Note

The transfer of requirements from a high level 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.



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.

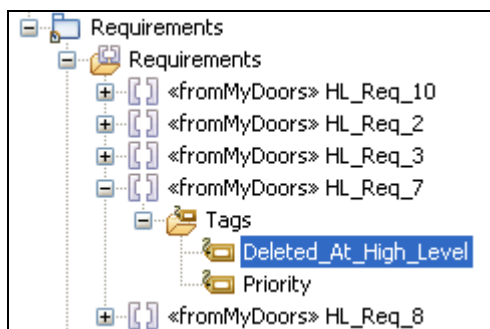
Modified Requirements

If requirements change, the modifications are automatically detected in Rhapsody Gateway, as described in details in the *User Manual*. 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 will not directly delete the requirements from the Rhapsody model. Rhapsody Gateway adds only a **Deleted at high level** tag on this kind of requirements into Rhapsody, as shown in the following picture.



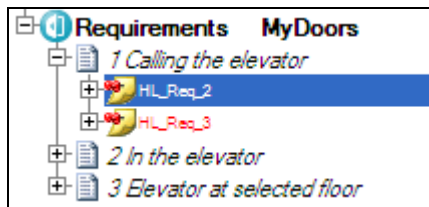
It is up to the Rhapsody user to delete the model elements according to the appropriate process.

Reaching a Requirement

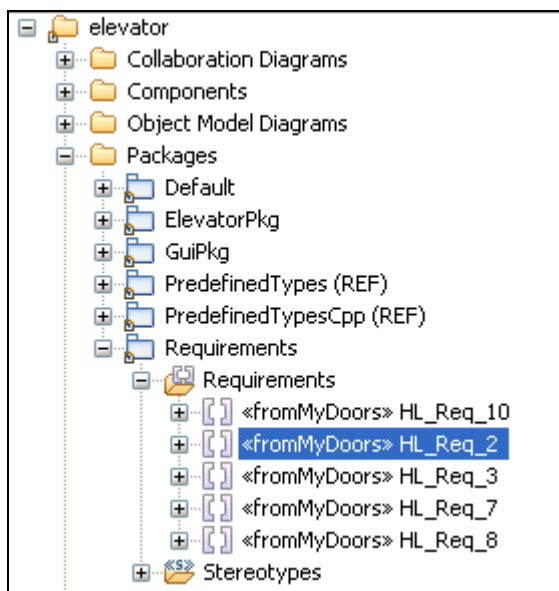
From Rhapsody Gateway, a specific navigation feature is available to reach a high level requirement in Rhapsody.

In order to perform the requirements underlining in Rhapsody, follow these steps:

1. From Rhapsody Gateway, select one requirement.



2. In the contextual menu, a new feature **Find in Rhapsody** is available. Click on this option.
3. In Rhapsody, the selected requirement is highlighted, such as follows:



Attributes / Rhapsody Tags Relationships

Tags are used to add information to the model base. Requirement attributes are managed using the powerful **Tag** capabilities of Rhapsody.

This section describes the attributes capture process.

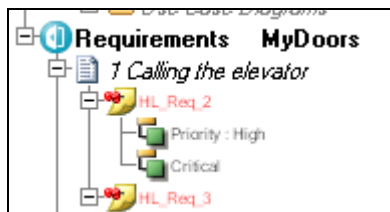
Representation of Requirements Attributes

Rhapsody Gateway attributes attached to captured requirements will be created as Rhapsody **Tags**.

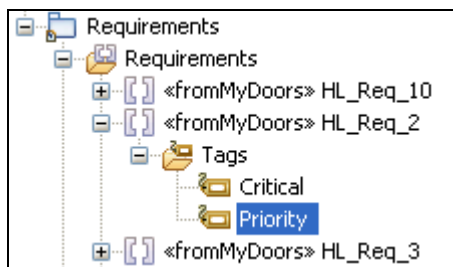
There are two types of attributes:

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

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



The corresponding representation in Rhapsody looks like the following example.



Capturing Rhapsody Tags

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

General Attributes Capture

Two types of tags can be defined in Rhapsody. **Local** tags are captured as attributes with their name and value. **Generic** tags are defined by a stereotype. These tags are available for a model element when the element is defined from this stereotype. But if the user overrides a tag value, the tag is considered to be local because the tag is tied to that specific element. With advanced capture, Rhapsody Gateway does as follows:

- ◆ Rhapsody Gateway will consider the generic tag and its value as an attribute if the generic definition is not changed by a local tag.
- ◆ If a local tag is defined from a generic one, Rhapsody Gateway will consider the local value for the element to be defined by a local tag.

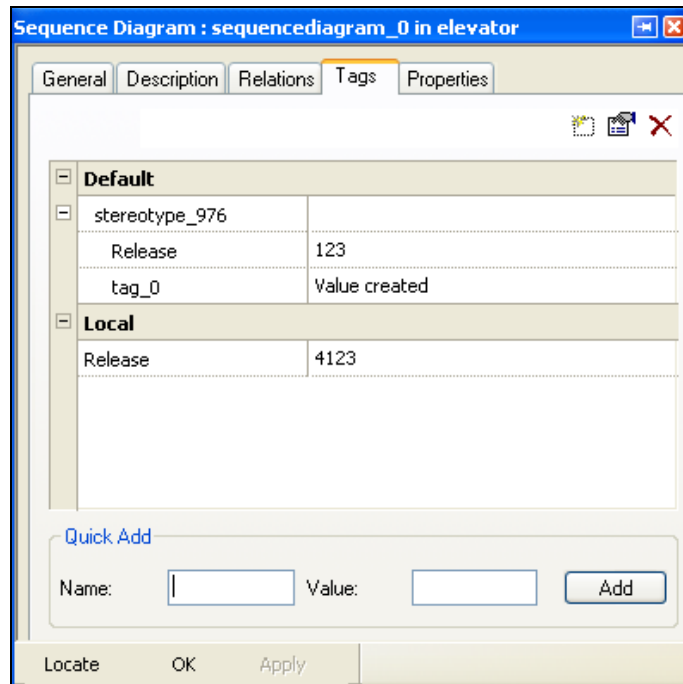
Below is an example:

1. Define a stereotype named **stereotype_976** which defines two tags:
 - ◆ Release = 123
 - ◆ Tag_0 = Value created... (it is a multi-lines value)
2. Create a sequence diagram named **sequencediagram_0** from the stereotype **stereotype_976**.
3. For this particular sequence diagram, replace the generic value by creating a local value 4123 for the Release tag.

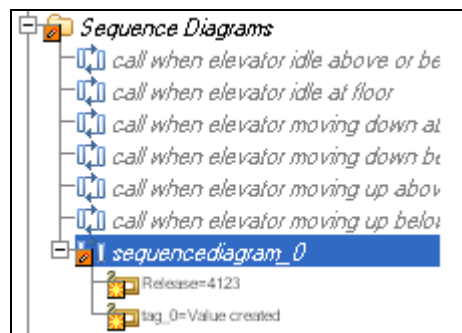
In this case, Rhapsody Gateway presents the sequence diagram with:

- Release = 4123 (local value)
- Tag_0 = Value created... (the multi-lines and generic value)

View of the Rhapsody tag definition



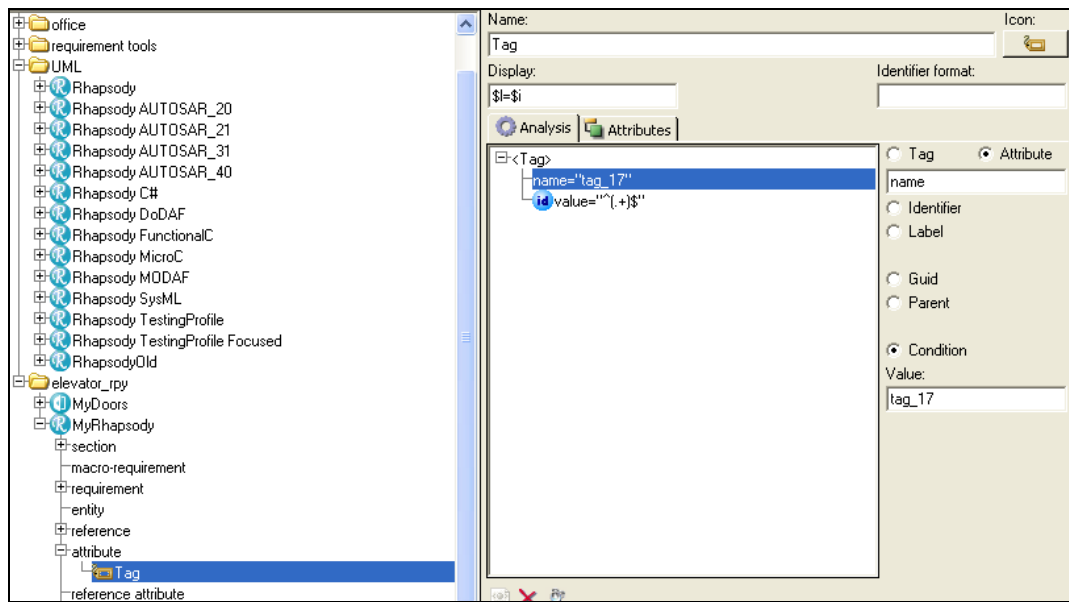
Corresponding Tag in Rhapsody Gateway



Capturing Specific Attributes

It is possible to capture only particular attributes from Rhapsody, or in other words, to modify the Rhapsody capture type. See the **Customizing Analysis Type** section of this document.

To capture specific tags, create an appropriate attribute and define a condition for the tags to be caught. With the definition of attributes presented in the following example, only tags named "tag_17" will be captured on the Rhapsody model.



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.

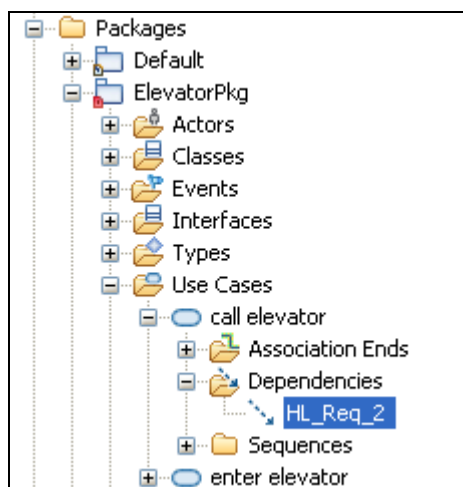
Using Dependencies

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.

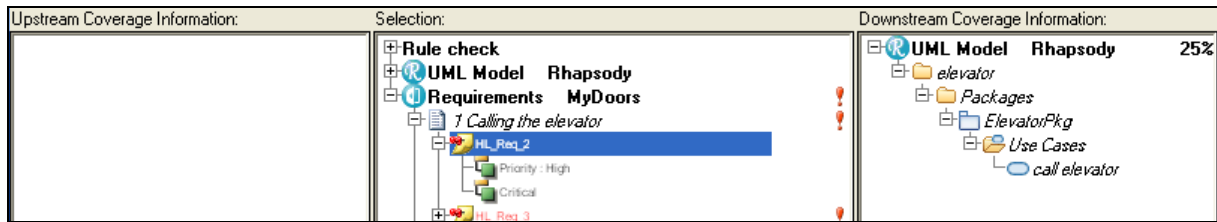
Dependencies with the following stereotypes: **trace**, **verify** and **satisfy** are captured as **References**, otherwise they are captured as **Links**. Refer to the Types editor to look at the default Rhapsody type properties.

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 more detailed information. See the *Rhapsody User Guide* for more information.

Using Rhapsody Anchors

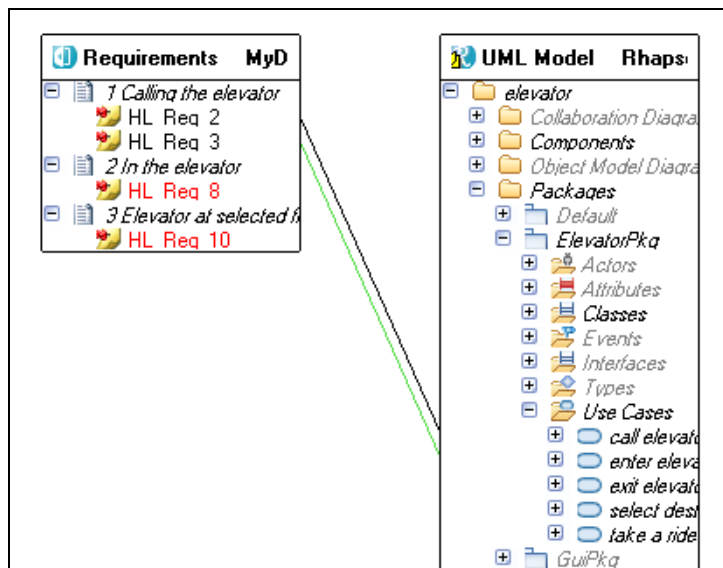
Rhapsody anchors creation is an old means of allowing the creation of traceability links; this is always useable. See the Advanced pane of the **Add high level** box.

See the *Rhapsody User Guide* for more information.

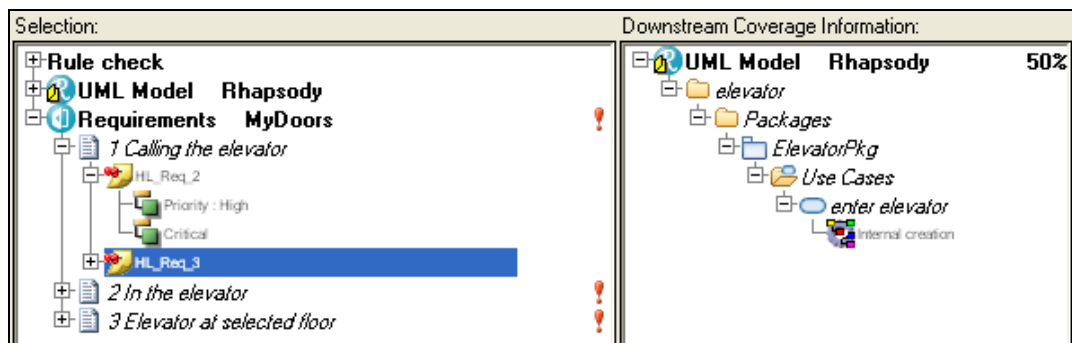
Links Created from Rhapsody Gateway

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

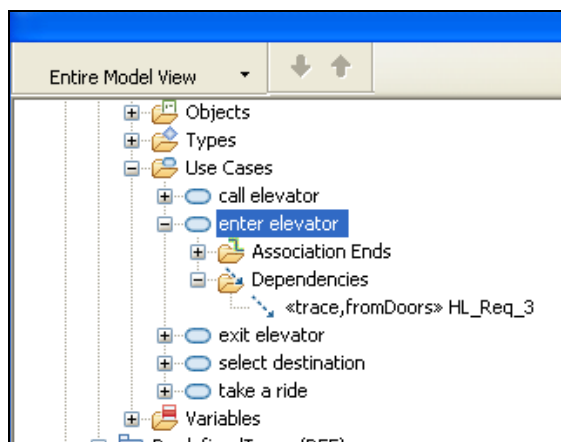
The following picture displays a created covering link. It is represented with a green line.



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



To insert externally created links into Rhapsody, launch the **Tools > Add high level requirements** operation. 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.



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 more information about DOORS-Rhapsody Gateway interface.

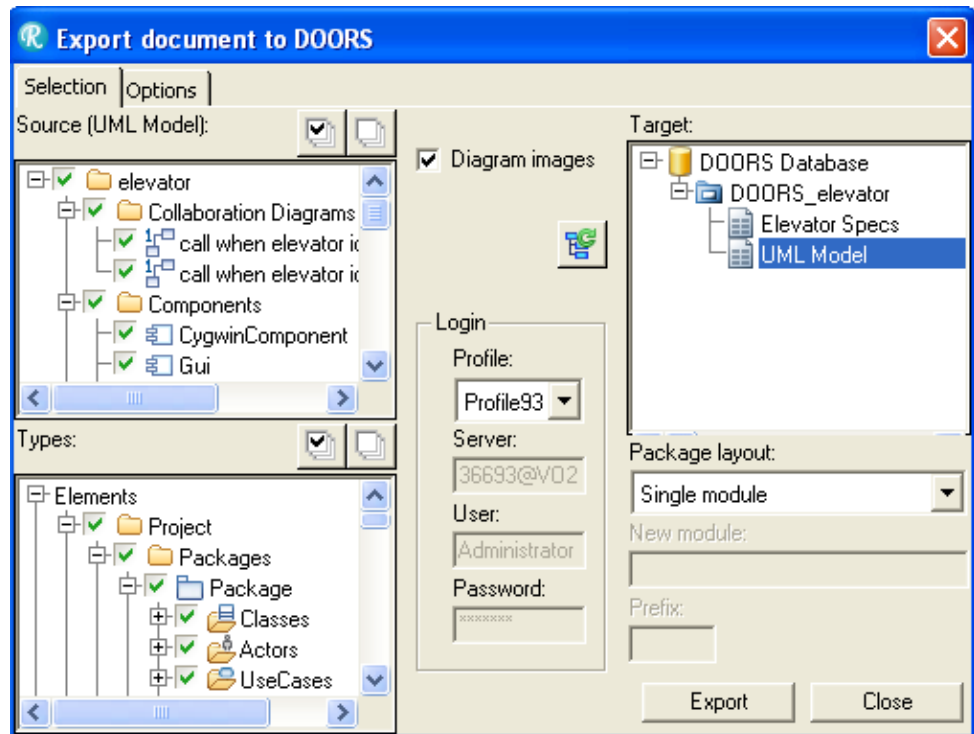
This section describes the upload process of elements into DOORS.

Creating Rhapsody Elements into DOORS

Rhapsody Gateway allows you to upload the model into DOORS with or without diagrams. Obviously, the diagrams need to be captured before being exported.

Follow these steps to perform the upload:

1. Select the UML Model from Rhapsody Gateway then choose **Export document to DOORS** option from the **Tools** menu. An Export dialog box opens.



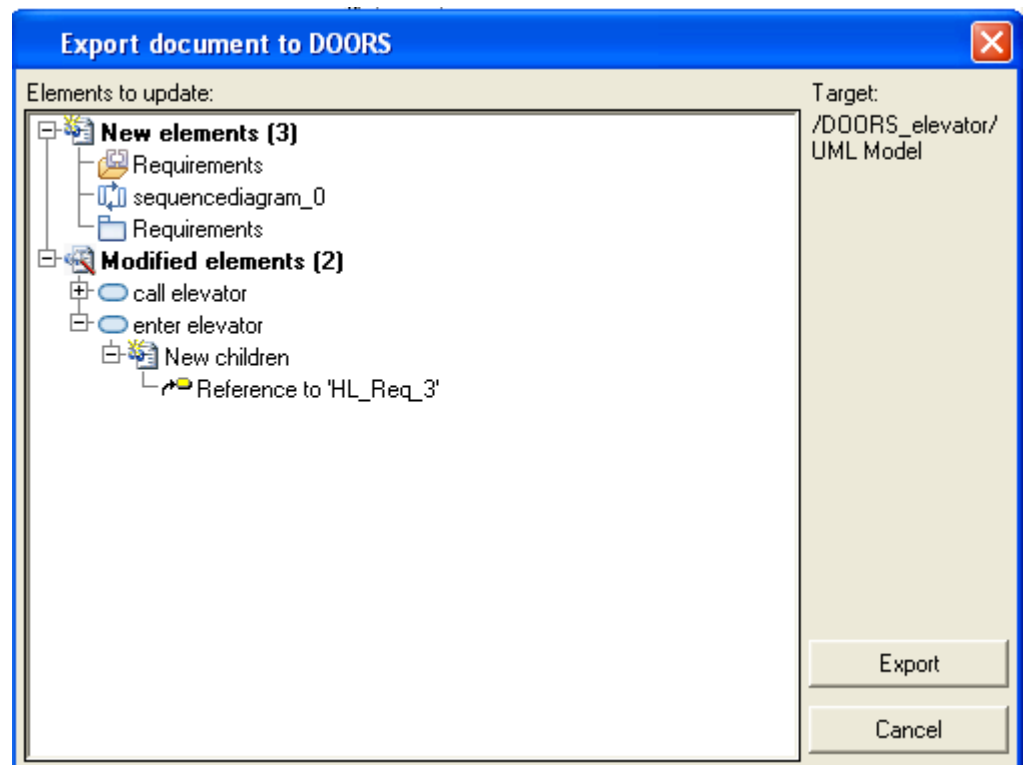
From this box, select the elements which have to be exported:

- ◆ The **Source** list on the upper left hand side of the window displays the levels of your current model hierarchy.
- ◆ The **Types** list displays a sub-selection of what was selected in the **Source** list. Within this subset, select the types that you want to upload.

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 will be 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 in order to check what needs to be updated. Then you will see an information window.



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. Use the **Cancel** option to close the window 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

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 in DOORS you change 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. The “Best Practice” is to 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 9c043223-8f29-43c9-9740-616b32a2d5b7

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.

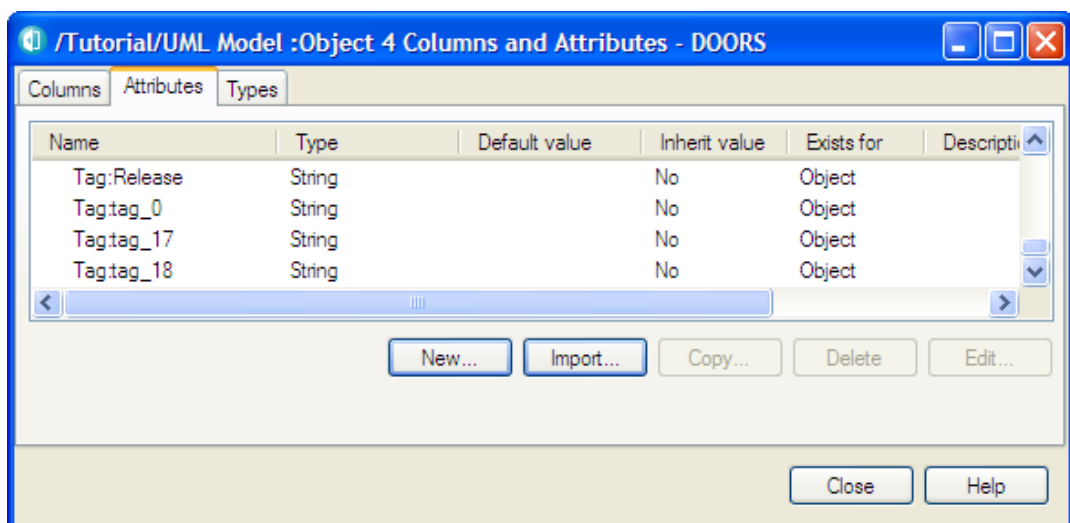
The Rhapsody **Tags** are created as DOORS **attributes**, as described in the next section.

Tags Management

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

For each new attribute a new column is created.

Each generic attribute is named Tag:<tag name>. Below is an example of a new attribute creation.



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



If some specific attributes are captured from Rhapsody Gateway, only one attribute is created. (See details in the section Capture of Specific Attributes for more information.) This tag is named **Tag**. The example below illustrates the addition of an attribute.



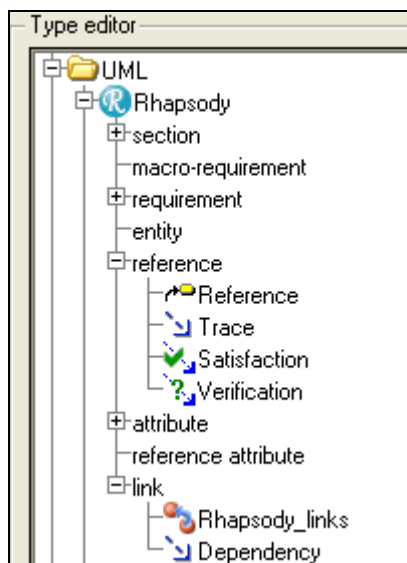
Rhapsody Gateway defines several ways to capture traceability information through **References** and **Links** in the type definition. See the *Customization Guide* and the *User Manual* to learn more about these concepts.

The default configuration for Rhapsody assumes:

- ◆ A definition of References will capture 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 will capture 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. See the following figure to compare the link creation.

Type definition in Rhapsody Gateway



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	

Export 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".

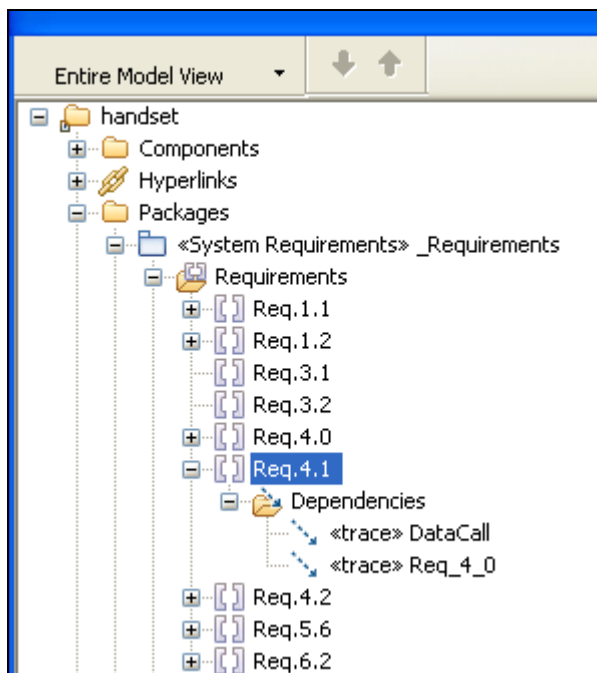
The example below comes from the Handset delivered in Rhapsody. As an example, the figure below shows the Rhapsody requirement Req. 4.1. This requirement contains two dependencies:

- ◆ 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.

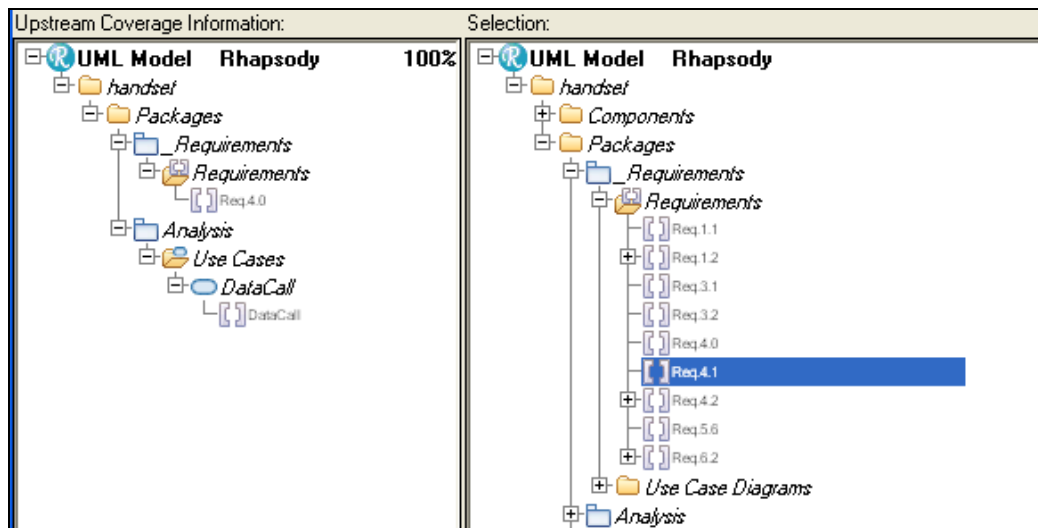
Change the stereotype of Req_4_0 into "trace".

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**.

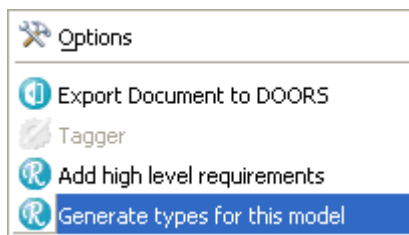
The concept of implicit requirements which are automatically created by Rhapsody Gateway also exists for DoDAF when the dependencies use the **Trace** stereotype.

Considering Other Profiles

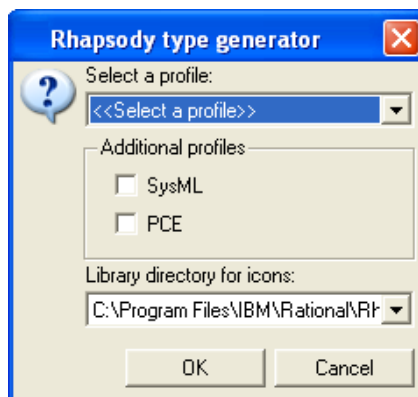
Rhapsody Gateway recognizes profiles. But all the profiles created by users cannot be found in Rhapsody Gateway as types. An option enables the generation of types which match the profiles applied to Rhapsody models, To better reflect information coming from Rhapsody.

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.



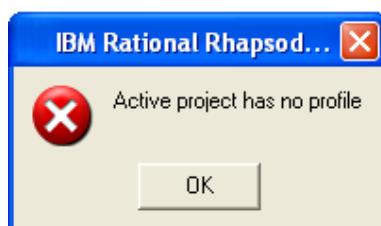
Select the main profile for the type from the combo-box.

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

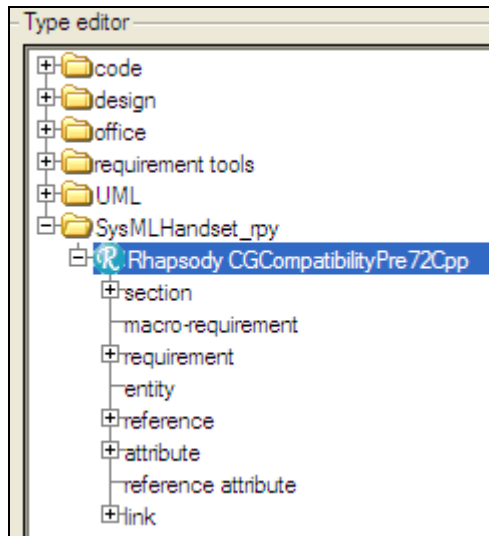
Select the directory to save icons coming from the profiles and which will be used in the new created type.

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.

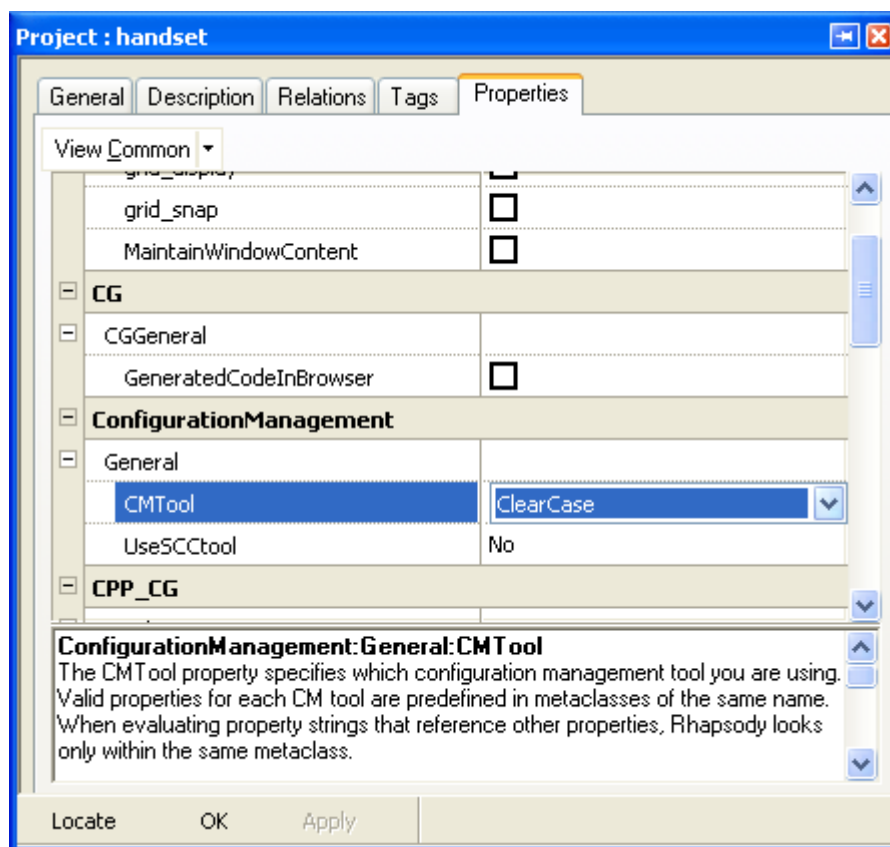


5. The corresponding Rhapsody document type in the Project configuration has automatically changed with the newly generated type. Launch the **Reload** feature on the model to re-execute the analysis of the model with the new type.

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.



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:

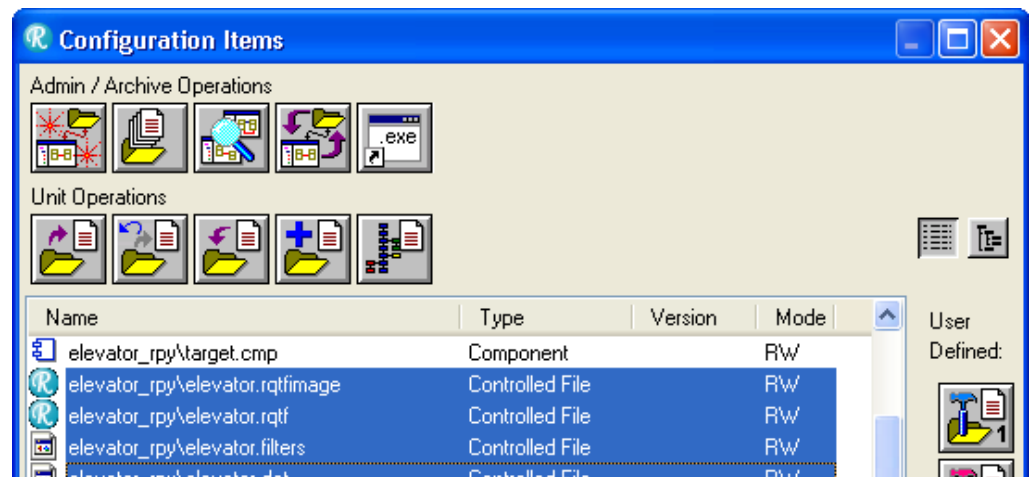
1. Create a Rhapsody Gateway project from Rhapsody. The Rhapsody Gateway project files are already saved in the `_rpy` sub-directory of your Rhapsody project directory.

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.



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.