<http://www.ibm.com/support/knowledgecenter/en/SSB2MU_8.1.4/com.ibm.rhp.doors.tutorial.doc/topics/intro_apply.html?view=embed>

# Managing requirements with Rhapsody Gateway and DOORS

You can import, analyze, and trace DOORS® requirements in the Rhapsody® Gateway, and to export Rhapsody model elements to DOORS.

The Rhapsody Gateway offers the following advantages:

* Enables you to see the upstream and downstream impact of requirement changes, in real time.
* Enables you to link requirements to model elements and analyze the coverage of the requirements. The impact of requirement changes can be viewed and analyzed.
* Enables you to exchange information with 3rd party authoring and requirements tools.
* Enables you to convert the input DOORS requirements to an intermediate file in a choice of formats, such as text or XML.

You can link requirements to model elements in Gateway, analyze the requirements, and the coverage of the requirements in the Rhapsody model

Traceability occurs when a requirement changes in DOORS.

You can upload Rhapsody model elements to DOORS, change a model element in Rhapsody, and export the modification to DOORS.

**Installation requirements for Rational DOORS with Rational Rhapsody**

To use the IBM® Rational® DOORS® interface:

* Copy the dxlapi.dll dynamic link library from the bin directory in the Rational DOORS installation to your winnt\system32 directory.
* Make sure that the file pc\_server.dxl is in the $OMROOT\etc directory of the IBM Rational Rhapsody® installation.
* Rational DOORS must be on the local machine.

Rational Rhapsody reads the locations of your Rational DOORS installation and license from the Windows registry and the LM\_LICENSE\_FILE environment variable. If for some reason Rational DOORS is not registered in the registry in the normal way and your LM\_LICENSE\_FILE variable is not set, you can manually set the InstallationDir and LmLicenseFile properties under RTInterface::DOORS. If these properties are set, they override the registry value or environment variable.

**To set the IBM® Rational® DOORS® properties in Rational Rhapsody:**

1. Click **File** > **Project Properties** and select RTInterface::DOORS.
2. Set the InstallationDir property to the location of your Rational DOORS installation. For example:  
     
   d:\doors
3. Set the LmLicenseFile property to the location of your Rational DOORS license. For example:  
     
   d:\doors\lib\license.dat  
     
   If you are using a client license for Rational DOORS, you now can enter a port number and server using the following format:  
     
   *<port>*@*<server>*

# Using Rational Rhapsody with Rational DOORS

## **Procedure**

1. Set up a project within Rational DOORS.
2. Capture requirements and other design information in Rational DOORS.

**Note:** Rational DOORS is the owner of the requirements. If you need to make changes to requirements, make them in Rational DOORS.

1. Capture the high-level use cases, structure, sequences, and behavior of the system in Rational Rhapsody.
2. Select elements from the Rational Rhapsody model that you want to trace to elements in Rational DOORS.
3. Export the selected elements as shadow copies to the Rational DOORS database.
4. Navigate to the exported elements in Rational DOORS from the Rational Rhapsody browser.
5. Create links within Rational DOORS between the requirements and shadow copies.
6. Run the Check tool to verify the consistency of shadows in the Rational DOORS database with elements in the Rational Rhapsody repository and the completeness of the links between the two.
7. Navigate from the Rational DOORS database to the respective elements in Rational Rhapsody, as wanted.

# Confirming and opening the Gateway from Rhapsody

The Rhapsody Gateway is an add-on product to Rhapsody that requires a separate license. Gateway operates in the following environments:

* Windows (NT4, 2000, XP)
* Unix (Solaris 2.6 and higher)
* Linux

To confirm that the Gateway is installed, and to work with Gateway:

## **Procedure**

1. Create, or open an existing C or C++ Rational® Rhapsody project.
2. To open the Rational Rhapsody Gateway, **Management View** window for a C or C++, select the project, right click n the model, and select **Rational Rhapsody Gateway** > **Open** from the drop down list. Click on the Management View tab, to open the view.
3. To open the Rhapsody Gateway Configuration window, select the project and click the converter converter button in the Rhapsody Gateway window

There are fundamentally two interfaces in Gateway, the main interface and the configuration dialog window. Each of these offers various views

# Importing project requirements into DOORS

You must import the requirements module into DOORS® prior importing them into Rhapsody® Gateway.

**Before you begin**

Before you begin, you must have administrator access to DOORS to import requirements module into DOORS.

The Elevator project is used as the requirements example. The project is in the Rhapsody/Gateway folder. The same model exists for both C++ and C programming languages; either one can be used for the exercise.

## **Procedure**

1. Start the DOORS client, click **View** > **Database View**.
2. For the *Elevator* example project, type *Doorselevatorpjt* in the **Name** field . The new *Doorselevatorpjt* folder appears listed under the **Doors Database**.
3. Select *Doorselevatorpjt* folder, and click from the bar menu, **File** > **Restore** > **Project**. The Restore Project - Doors window opens.
4. Click **Browse**, and select from the Program Files\IBM\Rational\Rhapsody\7.5.3\Gateway\examples\coupling\DOORS, the *Tutorial.dpa* file to load the *Elevator* requirements example. Click **OK**.
5. The *Elevator* project appears in the *Doorselevatorpjt* folder. Double-click the *Elevator Specs* file, for example, to open a requirements file. You can view the requirements in the Standard view.

Now, you can import the *Elevator Specs* requirements into a Rational® Rhapsody model, or add more requirements to the project before importing it.

# Importing project requirements into Rhapsody Gateway from DOORS

## **Before you begin**

In addition to Administrator access to DOORS®, you must have Rational® Rhapsody for C or C++, and the Rhapsody Gateway installed on your system for the import.

**Procedure**

1. In Rational Rhapsody for C or C++, click **File**> **Open** to load the elevator project from the Rational\Rhapsody\7.5.3\Gateway\examples\coupling\Rhapsody\Elevator install directory.
2. Select the example *elevator.rpy*. Right-click on the elevator project, and select **Rational Rhapsody Gateway** > **Open**.
3. The Rhapsody Gateway starts, and the Coverage Analysis View appears by default.
4. Click the **Edit Project** button converter, from the menu bar, to open the Rhapsody Gateway Configuration window.
5. Click the **Add Document** button converter in the middle of the page under the work area. Click to stamp the document into the main window. The default document name is **Document1**.
6. In the Rhapsody Gateway Configuration window, click on **Type of Analysis** tab to select **Doors Basic** from the dropdown list.

Note that the **Intermediate file** box is automatically checked off indicating that the intermediate file is saved.

1. Highlight the content in the **File or Directory** box, and click the browse **"…"** button that becomes functional as a result of the highlighting at the right of the tab. The Select DOORS module window opens where you are prompted to log into DOORS.
2. After you login into DOORS, select the **Update** button converter to display the Elevatorproject in the **DOORS Database** box of the Select DOORS module window
3. The Doorselelevatorpjt project appears listed under **DOORS Database** with all the Elevator Specs files underneath.

The DOORS requirements are now imported into the Gateway and the connection between the UML model and the requirements covered by the model is established.

1. To add a coverage link to the project, select one of the *specs* files, for example *Elevator Specs*, and click **OK**.
2. In the Rhapsody Gateway window, select the **Add Cover** button converter. Click **Apply** > **OK**.

**Note:** The direction of the arrow is important. The model always covers the requirement, so the direction of the arrow goes from the model to the requirement. The coverage link is now made; this establishes the connection between the model and the requirements.

1. Click **Yes** when the Project Analysis window appears prompting you to reanalyze the Rhapsody Gateway project.
   1. Return to the **Rhapsody Gateway Coverage Analysis View**, and confirm that here are two documents listed.
   2. Click on the requirements, and confirm the uncovered requirements that are highlighted in red converter.

**Note** that the sun symbol indicates the newly added elements.

# Importing requirements from Rhapsody into DOORS

To add requirements into Rhapsody® or DOORS® project, analyze the requirements, and the model elements covering the requirements.

## **Procedure**

1. In the Rhapsody Gateway window, select the **UML Model Rhapsody** project.
2. Click **Tools** > **Add high level requirements**, The Add high level requirements window opens.
3. In the Add high level requirements window, select the elevator project at the top level in the browser and click **OK**.
4. In the Rhapsody Gateway **Coverage Analysis View**window, select the *UML Model Rhapsody* project, and open the *Document1* folder to see the newly imported requirements.
5. Return to Rhapsody view, and select the *elevator Packages* folder to confirm that the requirements are listed with a type fromDoors Basic.

Now, the requirements are now successfully added into Rhapsody project. You can now associate requirements to model elements

# Linking requirements to Rational Rhapsody model elements

## **Procedure**

1. Open Rational Rhapsody, and load the same elevator project you have been using in this tutorial previously by opening elevator.rpy file.
2. Expand the **Use Case Diagrams** folder and locate the main uses diagram. Double-click the **main uses** diagram to open in the work area.
3. Select **Doors Requirement 2** in the browser and drag it onto the main uses diagram.
4. Right click the selected **Doors Requirement 2** on the **Use Case** diagram, and select **Notation Style** > **Box Style**.

Notice **Doors Requirement 2** on the use case diagram changes to a box with text information.

1. Click the Dependency button converter and drag the **Dependency** line from the **call elevator** use case to **Doors Requirement 2** on the diagram, and release it. The dependency link is shown.
2. Double-click the dependency line so the window shown below opens. Select trace from the Stereotype drop-down list, and click **OK**.
3. Double-click **Doors Requirement 2** on the diagram and view the requirement text.
4. Click **Doors Requirement 3** in the browser and drag it onto the main uses diagram. Right click the selected **Doors Requirement 3** on the Use Case diagram, and select **Notation Style** > **Box Style**.

Notice **Requirement 3** on the **Use Case** diagram changes to a box with text information.

1. Click the **Dependency** button.
2. Drag the dependency line from the enter elevator use case to **Requirement 3**on the diagram and release it. The link is shown. Double-click the dependency line so the dependency dialog box opens. Select tracefrom the Stereotype dropdown list, and click **OK**.
3. Save the changes, and right click on the elevator project and select **Rhapsody Gateway**to start the Gateway.

Select **Yes** when asked **Do you want to reanalyze the project?**. Take note of the orange symbols that indicate modifications that have occurred.

1. Select **Coverage Analysis View**, and in the middle **Selection** column, expand the **Document1 Doors XML** folder and view the requirements listed. Confirm **Doors Requirement 2** and **Doors Requirement 3** that have now turned to black color indicating they have coverage. Under the **UML Model Rhapsody** in the center **Selection** column select the elevator project. In the left **Upstream Coverage** column confirm the **Req 2** and **Req 3** appear indicating the requirements are covered by elevator model elements.
2. Click **Req 2** in the middle **Selection** column. Confirm that in the right **Downstream Coverage** column, the call elevator use case is shown indicating it provides coverage for **Req 2**. Click **Req 3** in the middle **Selection** column. Confirm that in the right **Downstream Coverage** column, then the enter elevator use case is shown indicating it provides coverage for **Req 3**.

Once the link is made between a requirement and a model element, the Rhapsody Gateway provides analytic information. The covered requirements appear in black color, and uncovered requirements appear in red color.

The percentage of total DOORS® requirements covered by the Rational Rhapsody model elements is indicated in Rhapsody Gateway window.

# Adding new requirements with DOORS wizard

## **Procedure**

1. In Rhapsody Gateway window, select the elevator project, and click the converter button to open the Rhapsody Gateway Configuration window.
2. Click the **Add Document** button converter in the middle of the page under the work area. Click to stamp the document into the main window. The default document name is **Document1**.
3. In the Rhapsody Gateway Configuration window, click on **Type of Analysis** tab to select **Doors Basic** from the dropdown list.

**Note** that the **Intermediate file** box is automatically checked off indicating that the intermediate file is saved.

1. Highlight the content in the **File or Directory** box, and click the browse **"…"** button that becomes functional as a result of the highlighting at the right of the tab. The Select DOORS module window opens where you are prompted to log into DOORS.
2. In the Configuration window, select **Document1**, right-click and select **DOORS Wizard**.
3. The Type Customization window opens. Select the Elevator Specs file, for example, to which to add a new requirement and click **Next**. In the next opened Type Customization window, select **Doors Basic**, for example, in the **Select a DOORS Type** drop down field. Check the **Create a new type** box, type the name of your new requirement, and click **Next**.
4. Select the Requirement value in the Kind column from the dropdown list of values, and click **Next** > **Finish**. The Type Customization window closes.
5. Select the Requirement value in the Kind column from the dropdown list of values, and click **Next** > **Finish**. The Type Customization window closes.
6. In the Configuration window, click **Apply**> **OK**. Answer **Yes** to the message window that asks you to reanalyze the project. The requirement is imported into DOORS.
7. In Rhapsody Gateway window, **Coverage Analysis View**> **Selection** field, expand **Rule Check** > **Uncovered requirements**.
8. Double click on the uncovered requirement to open the *Elevator specs* requirements file in DOORS, and check that the requirements are listed.
9. In Rhapsody Gateway window, select **UML Model Rhapsody** and click **Tools** > **Add high level requirement**. The Add high level requirements window opens.
10. In the Add high level requirements window, select the *elevator* folder as the root package for the requirement, and click **OK**. The Add level requirements conformation window opens after the requirement is saved.
11. Click **OK**, and **Yes** when you are asked of you want to reload the model in Rational Rhapsody.
12. In Rational Rhapsody window, open the **Documents1** > **Requirements** to check if the new requirement was added to the elevator project.

Now, you can add the new requirement to one of your use cases by opening the use case in Rational Rhapsody window. From the **Documents1** > **Requirements**, you can drag the new requirement onto the use case diagram. You can also add new relations to the requirement, such as dependencies, links, and so on.

# Updating requirements in DOORS

## **Procedure**

1. Start Rhapsody and load the same elevator project you have been using in this tutorial by opening the elevator.rpy file. Right click elevator and select Rhapsody Gateway to open the **Coverage Analysis View**.
2. Double-click **HL\_REQ 2** in the left column under **Document 1 Doors XML**, enter your DOORS login, and click**OK** to open DOORS.
3. Edit the requirement text for **HL\_REQ 2** in the **Selection Text** box, for example, by adding the word **down** after **or**. Save the changes to the DOORS requirement.
4. Reopen the Rational Rhapsody Gateway window and select **Document1 Doors XML** in the left column of the **Coverage Analysis View**.
5. Right click **Document1 Doors XML** to open a dialog, and select **Reload** for DOORS to run on the reload. Confirm the orange box on **HL\_REQ 2** appears indicating a change has occurred.
6. Select the **Coverage Analysis** view tab, and click **HL\_REQ 2** in the middle **Selection**column. Check the text from the **Selection Text** box below and confirm that the requirement *down*is added to the text
7. To add a new requirement, in the Management View, select**Tools** > **Add high level requirements** from the menu bar.
8. When prompted to reload the UML Model select **Yes**.

Note the orange symbol indicating the UML Model has changed, and the orange symbol next to the main uses **Use Case** model.

1. Return to the open Rhapsody window and double-click on **HL\_REQ 2** in the main uses **Use Case** diagram. View the requirement text in the dialog in the **Specification**section, and check that *down* was added to the text.

The change to a requirement was made in both Rhapsody and Gateway. Now, you can export Rational Rhapsody model elements to Doors, and view those elements in DOORS.

# Exporting Rational Rhapsody model elements

## **Procedure**

1. Start Rational Rhapsody and load the same elevator project you have been using in the previous lesson by opening *elevator.rpy* file.
2. Select the elevator project, right click elevator and select **Rhapsody Gateway**. The Gateway Management window opens.
3. In the **Coverage Analysis** view, select **UML Model Rhapsody** in the center **Selection** column.
4. In the Rhapsody Gateway option, select **Tools** > **Export Elements to DOORS** to open **Export elements to DOORS** window.

The left hand part of the dialog allows the selection of what to export:

* + The **Source UML Model** list allows you to select objects individually
  + The **Types** list allows you to filter the UML element types that are uploaded into DOORS

The right part of the dialog provides a tree view of the DOORS database. The tree is viewable when you click on the update button converter.

If the diagram images box is checked, images will also be uploaded to DOORS.

1. In the **Source UML Model** box on the top left, check only the **ElevatorPkg** box, clear the diagram images box,.and login to DOORS.
2. In the **New module** box on the bottom right enter the name **UML elevator model**.
3. Click the **Update** button to view the DOORS database tree, and select the **DOORSelevatorpjt** file in the tree. Click **Export**

When you click on **Export**, the synchronization between Gateway and DOORS starts to perform the export.

1. Check to see what is being exported, and confirm the export by clicking the **Export** button. A different **Export elements to DOORS** dialog opens that says that DOORS export operation is completed.
2. Click the **OK** button and click **Close** in the original **Export elements to DOORS** dialog.

All model elements are successfully exported to DOORS.

# Changing an element in Rhapsody to export to DOORS

When a change is made to a model element that has been exported to DOORS®, the model element must be reexported. Rhapsody® Gateway must be reloaded because the UML Model changed. The Export to DOORS option is selected from the Gateway and the modified model element is exported. Modifications are easy to spot in Gateway because they are always indicated with the orange symbol.

## **Procedure**

1. Start Rational® Rhapsody and load the same elevator project you have been using in this course previously by opening elevator.rpy file. Open the main use case diagram.
2. Double-click on the call elevator use case to open the following window.
3. In the **Name** field rename call elevator to call an elevator. Click **OK**, and save.
4. Select the elevator project, right click and select **Rhapsody Gateway** from the list. When the Rhapsody Gateway elevator opens, click **Yes** to update the project in Rhapsody Gateway.

When the Rhapsody Gateway Management window opens, confirm the orange symbols appear on the **UML Model Rhapsody** in various places. In the browser under **ElevatorPkg** > **Use Cases** confirm the use case call elevator is now call an elevator.

1. On the Coverage analysis view, in the Selection column, select the **UML Model Rhapsody** model.
2. Select **Tools** > **Export Elements to DOORS** from the top menu. The Export Elements to DOORS dialog opens.

Be sure only the ElevatorPkg is checked off in the left **Source** column, and the diagram images box is not checked.

1. In the **Target** box on the right top, select the **UML elevator model**, so the element is exported there. Click **Export**.
2. A dialog appears showing **Elements to update**. Confirm call an elevator use case appears, and click the **Export**button.
3. When **Doors export operation completed** message appears, click **OK** and **Close**.

The changes to the model are now exported to DOORS.